2020

Towards Ethical Machine Learning: New Algorithms For Fairness And Privacy

Seth Neel
University of Pennsylvania

Towards Ethical Machine Learning: New Algorithms For Fairness And Privacy

Abstract

The challenge of ensuring that tools for data science and machine learning enforce ethical notions like privacy and fairness is one of the most important facing modern computer scientists. While the last decade has seen a flurry of research in this area, there are still significant challenges to using existing algorithms and definitions in practice. This thesis considers the theoretical questions arising from practical considerations, with an emphasis on machine learning applications. In particular, we make crucial definitions and obtain new results towards answering the following questions:

• How can we learn optimally private classifiers subject to a hard accuracy constraint?

• How can we leverage heuristic optimization oracles for private learning while still maintaining rigorous privacy guarantees?

• How can we extend the coarse fairness protections provided by statistical notions of fairness to richer subgroup classes?

• How can we learn subject to an individual fairness notion whose metric is not provided, but is instead learned from a panel of experts? Behavioral subject experiments validate theoretical results.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Statistics

First Advisor

Michael J. Kearns

Second Advisor

Aaron L. Roth

Keywords

Computer Science, Data Privacy, Fairness, Game Theory, Machine Learning, Online Learning

Subject Categories

Artificial Intelligence and Robotics

TOWARDS ETHICAL MACHINE LEARNING:

NEW ALGORITHMS FOR FAIRNESS AND PRIVACY

Seth V. Neel

A DISSERTATION

in

Statistics

For the Graduate Group in Managerial Science and Applied Economics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

2020

Supervisor of Dissertation

_____

Michael J. Kearns

National Center Professor of CIS

Co-Supervisor of Dissertation

_____

Aaron L. Roth

Class of 1940 Associate Professor of CIS

Graduate Group Chairperson

_____

Nancy Zhang, Ge Li and Ning Zhao Professor, Professor of Statistics

Dissertation Committee:

Weijie Su, Assistant Professor of Statistics
Bhaswar Bhattacharya, Assistant Professor of Statistics

TOWARDS ETHICAL MACHINE LEARNING:

NEW ALGORITHMS FOR FAIRNESS AND PRIVACY

DEDICATION

To my family:

Dylan, for signing on for the adventure.

Dad, for raising me to believe in the power of ideas.

Mom, for raising me to believe in myself.

ACKNOWLEDGEMENT

First and foremost I'd like to thank my advisors, Aaron and Michael. Thank you for teaching me everything I know about being a computer scientist, I'll always try to emulate your examples in my own career. Your infectious appetite for new ideas teaches by example, without the need for articulation. Thank you for being my biggest advocates, trusted mentors, brilliant collaborators, and friends. I look forward to many fruitful years of collaboration and friendship ahead.

I'd also like to thank my stellar collaborators and co-authors during my Ph.D. In particular, my good friend Steven Wu, for some amazing contributions to our research projects, advice, and good times in general. Also, Jamie Morgenstern, Matthew Joseph, Adam Smith, Hadi Elzayn, Chris Jung, Giuseppe Vietri, Katrina Ligett, and Jieming Mao.

ABSTRACT

TOWARDS ETHICAL MACHINE LEARNING:
NEW ALGORITHMS FOR FAIRNESS AND PRIVACY

Seth Neel

Michael J. Kearns

Aaron L. Roth

The challenge of ensuring that tools for data science and machine learning enforce ethical notions like privacy and fairness is one of the most important facing modern computer scientists. While the last decade has seen a flurry of research in this area, there are still significant challenges to using existing algorithms and definitions in practice. This thesis considers the theoretical questions arising from practical considerations, with an emphasis on machine learning applications. In particular, we make crucial definitions and obtain new results towards answering the following questions:

- How can we learn optimally private classifiers subject to a hard accuracy constraint?
- How can we leverage heuristic optimization oracles for private learning while still maintaining rigorous privacy guarantees?
- How can we extend the coarse fairness protections provided by statistical notions of fairness to richer subgroup classes?
- How can we learn subject to an individual fairness notion whose metric is not provided, but is instead learned from a panel of experts? Behavioral subject experiments validate theoretical results.

Contents

List of Figures

CHAPTER 1

Introduction

1.1. Towards Ethical Machine Learning

Buoyed by the rise of vast technology platforms that collect troves of our most sensitive data, the digital transformations upending even the most entrenched legacy industries, and breakthroughs in computing, in the last decade machine learning has become increasingly ingrained in the services that power modern business and daily life. Unlike mere automation, machine learning promises to do things not only more efficiently, but more intelligently. Dozens of startups and Fortune 500s alike have promised to make more accurate loans, tailored advertisements, recidivism decisions, and even beauty contest judgements Levin (2016). In addition to achieving greater accuracy, at the outset machine learning models also seemed to hold the promise to be more *ethical*. After all, by definition they replace the inconsistent, possibly prejudicial decisions of humans with the certainty of mathematics. As recently as 2019 conservative columnist Ryan Saavedra ignited a twitter war with Congresswoman Alexandria Ocasio-Cortez over whether algorithms could be racist, and the Trump administration has sought to pass housing legislation that protects algorithm-driven decisions by landlords from any accusations of discrimination Dane (2019).

Yet in almost every domain where machine learning has been used to make consequential decisions about humans, there have been documented cases of algorithmic misbehavior.

These instances of misbehavior are typically of two distinct types: trespasses against users whose data was input to the algorithm, or adverse outcomes for individuals who are effected by decisions made by the system. In machine learning parlance, this is a distinction between users in the training set, and those effected by the model post deployment.

Violations of this first kind, for users in the training set, are a form of *privacy* violation – as a result of the witting or unwitting inclusion of their data in the system, private user data has been exposed. The use of sophisticated algorithms, and the explosion of new data sources, has been shown in several highly publicized instances to enable re-identification of data subjects, even when all the released data has been "anonymized". The most prominent instance of this was the re-identification attack on the Netflix dataset of Narayanan and Shmatikov (2008), where anonymized Netflix data was combined with public IMDB data, and used to re-identify the supposedly anonymous accounts in the Netflix dataset. As a result, algorithm designers in the public and private sectors have a moral imperative, and in many instances a legal one, to balance the utility of their products with guaranteeing meaningful privacy protections for their users.

Violations of this second kind, perpetrated against the end users in the system, are typically issues of *fairness*. Whether due to historical bias present in the training data, sample size disparities for different sensitive groups, redlining, over even deliberate discrimination, there exists a protected class of end users who have been significantly harmed relative to other groups by the algorithm. A particularly impactful example, both for the high-stakes domain it concerned, and the ensuing publicity and impact it generated, is the 2016 ProPublica article *Machine Bias*. In it they highlight how the algorithm developed by for-profit company Northpointe for recidivism prediction in Broward County, Florida was biased against blacks Angwin et al. (2016). In the last few years well-documented cases and think pieces on algorithmic discrimination have appeared almost weekly in the leading news outlets Rudin (2013); Miller (2015a,b), along with a concurrent explosion

of interdisciplinary research in computer science, philosophy, law, and statistics aimed at rectifying and codifying bias in algorithms.

Given the flagrant abuses of privacy and fairness discovered in algorithmic decision-making over the past few years, it can be tempting to condemn algorithmic decision making as a whole. Indeed, there may be situations in which it is inappropriate for an algorithm to replace a human being, for a multitude of reasons. That being said, the perspective of this researcher, is that in most cases it is far more wise to design a better algorithm, than to forsake the endeavor. After all, human decision makers exhibit precisely the same proclivities to privacy disclosure and discrimination as algorithms, only in the case of the latter, it is precisely their algorithmic nature that makes it possible to codify and constrain these behaviors. The work of designing more fair and private algorithms therefore is the work of a techno-optimist; a journey across a multitude of different topics in statistics and machine learning with the goal of making them more private, fair, and ultimately ethical.

What follows in this thesis is a small but early contribution to this growing field of research, focusing on the notions of privacy and fairness, with a special emphasis on the theoretical questions arising from their incorporation into the practice of machine learning.

1.2. Differential Privacy

Many notions of privacy have been proposed in the computer science literature. The most obvious measure is to remove unique personal identifiers from the dataset. This has been shown to be woefully inadequate, since attributes that are not obvious unique identifiers can be used to infer sensitive attributes, particularly when they are combined across multiple datasets (a procedure known as “linking”). See Davis and Osoba (2016) for a thorough discussion. A variant of this scheme, termed “syntactic anonymization" seeks to modify the non-sensitive attributes of the dataset so that the dataset has many indistinguishable rows - the reasoning being that if k rows are indistinguishable, privacy will be preserved since each individual’s information is attributed to the group it is a member of rather than

the individual itself. Syntactic anonymization schemes include k-anonymity, and variants l-diversity, and t-closeness, all of which have been shown to suffer major shortcomings against *attribute disclosure attacks*. The intuition behind an attribute disclosure attack is that notions like k-anonymity do not prevent an adversary learning which group of k-records an individual belongs to, and if the distribution of sensitive attributes in that group is different than the overall base rate, group membership can leak sensitive information. Moreover, variants like t-closeness which try to equalization the group-wise distributions of sensitive attributes necessarily destroy utility in many settings, and so this appears to be a fundamental problem with the approach. For a more robust discussion we refer to Domingo-Ferrer and Torra (2008).

Differential privacy, introduced in the seminal paper of Dwork et al. (2006a), marked a major turning point in the science of private statistical disclosure. Differential privacy avoids the specific vulnerabilities discussed above, and provides rigorous privacy guarantees even in the face of adversaries with arbitrary prior information. While we give a formal definition in Section , we briefly motivate differential privacy and build intuition about the privacy protection it offers to individuals. We then summarize a few basic areas of research in differential privacy and applications of differential privacy outside of the privacy realm.

Differential privacy is a property of an algorithm $\mathcal{A}$ (or sequence of algorithms) operating on data, rather than a property of a specific output. Thus we wouldn't determine whether a specific data analysis result was private, but rather whether the computation that produced the result itself was private, independent of the actual output. Indeed Wood et al. (2018) remarks that "research has continually demonstrated that privacy measures that treat privacy as a property of the output, such as k-anonymity and other traditional statistical disclosure limitation techniques, will fail to protect privacy," and illustrates this point with several examples. Informally, we say that a procedure is differentially private, if any potential adversary can not learn much more about an individual in the dataset's private data based on the output of the study, than they could have inferred based on

the output of the study if the individual did not participate in the study at all. Note that this is different than saying that the output of the study doesn't cause them to learn more about the individuals private data, than if the study hadn't been published at all. In the words of Wood et al. (2018), "Risks of this nature apply to everyone, regardless of whether they shared personal data through the study or not," what differential privacy mitigates is the incentive of an individual to not participate or to lie in the study based on privacy considerations.

Differential privacy achieves this property by injecting random noise into typically deterministic computations, that serve to mask the contributions of individual data points to the outcome. Determining the precise scale of the added noise, the manner in which it is injected, and the subsequent privacy-accuracy analysis is a research question that can be applied to nearly any computation. This started as privatizing simple computations like releasing aggregate statistics, histograms, and maximizers, to more complex statistical tasks like linear regression, hypothesis testing, inference, and PCA, to state of the art objectives in machine learning: clustering, empirical risk minimization, and deep learning. Due to both the rigorous privacy guarantee differential privacy provides, and as differential privacy gains increasing acceptance in the legal and commercial spheres, this makes the study of differential privacy an increasingly compelling lens to tackle interesting and important research problems across statistics and computer science.

While in this thesis we use either the original definition of differential privacy, called $(\varepsilon, 0)$-DP, or its relaxation (ε, δ)-differential privacy, over the last few years various definitions of differential privacy have cropped up that seek to rectify problems with the original definitions; in particular to give tighter privacy analysis of common DP mechanisms like the Gaussian Mechanism, or tighter analysis of the composition of multiple mechanisms. As early alternatives have cropped up, further definitions have been proposed to mitigate the shortcomings of these definitions themselves. Nevertheless, the original notions of differential privacy remain the most prevalent in both research, and in industry deploy-

ments . The formal notion of differential privacy is a statement that the *max divergence* between the output distribution of $\mathcal{A}$ run on a dataset d, and the output distribution on any neighboring dataset d' is bounded by ε. New definitions also bound some distance measure between these two output distributions, but do so in subtly different ways. These include Concentrated DP (CDP) which imposes a sub-Gaussianity condition on the privacy loss random variable Dwork and Rothblum (2016), Renyi DP based on the Renyi divergence that is a relaxation of CDP and satisfies group privacy Mironov (2017), and the very recent f-DP which formulates privacy in terms of hypothesis testing between Gaussian distributions Dong et al. (2019). Joint differential privacy Hsu et al. (2013a) is a relaxed notion of differential privacy tailored to allocation problems, which assumes that the outputs to the rest of the users is private as a function of the input of a given user. This guarantees that by participating in the allocation problem, even if the other users collude against you, they can't infer your private data from their allocations.

Differential privacy is a strong stability property of an algorithm, and as a concept has found application for uses other than preserving privacy. One fascinating line of work explores the connection between differential privacy and what is called *adaptive data analysis*. In adaptive data analysis, an analyst asks repeated queries against the same dataset d, using answers to previous queries to formulate the next query. Many common routines fall under this framework, for example, stepwise regression or hyperparameter tuning of an ML algorithm. The goal of the mechanism answering the queries is to provide answers that are approximately accurate, and prevent the analyst from asking a query that "overfits" the dataset. Alternatives to differential privacy as a stability notion for adaptive data analysis include KL-stability and compression bounds, with differential privacy attaining the optimal rates Dwork et al. (2015a). Work by this author not included in this thesis extends these techniques to data gathered adaptively by bandit algorithms Neel and Roth (2018), and uses a new simplified proof technique to obtain the best known bounds for adaptively answering a sequence of low sensitivity queries . Joint differential privacy has also found use as a technique to incentive truthfulness in allocation problems, beginning

with the work of Hsu et al. (2013a), and applied in work from this author Jung et al. (2019) to the problem of private securities lending.

The privacy chapters included in this thesis focus on private machine learning – the task of approximately finding a hypothesis that minimizes error on a given dataset (ERM), subject to differential privacy. There is a rich and substantial literature on private convex ERM, weaving tight connections between standard mechanisms in differential privacy and standard tools for empirical risk minimization. The proposed methods for private ERM include output and objective perturbation (Chaudhuri et al., 2011; Kifer et al., 2012; Rubinstein et al., 2009; Chaudhuri and Monteleoni, 2008), covariance perturbation (Smith et al., 2017), the exponential mechanism (McSherry and Talwar, 2007; Bassily et al., 2014), and stochastic gradient descent (Bassily et al., 2014; Williams and McSherry, 2010; Jain et al., 2012; Duchi et al., 2013; Song et al., 2013).

Chapter 3 observes: while the literature above deals with problem of finding the optimally accurate classifier subject to a fixed privacy budget ε, the practitioner may have a hard accuracy constraint α, and simply view privacy as a non-essential desiderata subject to that accuracy constraint. How can we learn the optimally private classifier subject to a fixed accuracy constraint? Chapter 4 starts from the observation that while computational hardness results pervade any realistic machine learning setting, the successful use of learning heuristics (SGD, convex relaxations, etc.) has allowed the field to flourish. When learning under the constraint of differential privacy, can we use these same heuristics in a black-box way, without sacrificing rigorous privacy guarantees?

1.3. Fairness

As machine learning is being deployed in increasingly consequential domains (including policing (Rudin, 2013), criminal sentencing (Barry-Jester et al., 2015), and lending (Koren, 2016)), the problem of ensuring that learned models are *fair* has become urgent.

Prior to the recent focus on fairness in the machine learning literature, fairness has been a long studied consideration in philosophy, law, and economics. The most prevalent notion, one which has been codified in various landmark pieces of legislation (Civil Rights Act, Title VII, Equal Employment Opportunities Act) is known as *disparate impact*. Barocas and Selbst (2016) state:

Disparate impact refers to policies or practices that are facially neutral but have a disproportionately adverse impact on protected classes. Disparate impact is not concerned with the intent or motive for a policy; where it applies, the doctrine first asks whether there is a disparate impact on members of a protected class, then whether there is some business justification for that impact, and finally, whether there were less discriminatory means of achieving the same result.

Even if we agree that disparate impact is a reasonable criterion for discrimination, there is substantial flexibility on how to translate it into mathematics. In the past few years many different formulations have been proposed: equality of opportunity Hardt et al. (2016), demographic parity, predictive rate parity Hébert-Johnson et al. (2017), counterfactual fairness Kusner et al. (2017), and individual fairness Dwork et al. (2012) to name a few. Moreover, it has been shown that many of these definitions are themselves incompatible, requiring a careful domain-specific chose of fairness metric for any given task Kleinberg et al. (2016).

That being said, approaches to fairness in machine learning can coarsely be divided into two kinds: *statistical* and *individual* notions of fairness. Statistical notions typically fix a small number of protected demographic groups $\mathcal{G}$ (such as racial groups), and then ask for (approximate) parity of some statistical measure across all of these groups.

Definition 1.3.1. *Fix a collection of "protected groups" $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_k\}$. Given k distributions $\mathcal{D}_i$ (one for each group) over feature/label pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and a pair of functions on label pairs $s, g : \mathcal{Y} \times \mathcal{Y} \to \{0, 1\}$, a classifier A is ε-statistically fair with respect to (s, g) and $\mathcal{G}$ if for each*

pair of groups i, j:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_i}[s(A(x),y)|g(A(x),y)=1] \leq \mathbb{E}_{(x,y)\sim\mathcal{D}_j}[s(A(x),y)|g(A(x),y)=1] + \varepsilon$$

One popular statistical measure asks for equality of false positive or negative rates across all groups in $\mathcal{G}$ (this is also sometimes referred to as an *equal opportunity* constraint (Hardt et al., 2016)). Another asks for equality of classification rates (also known as *statistical parity*). These statistical notions of fairness are the kinds of fairness definitions most common in the literature (see e.g. Kamiran and Calders (2012); Hajian and Domingo-Ferrer (2013); Kleinberg et al. (2017); Hardt et al. (2016); Friedler et al. (2016); Zafar et al. (2017); Chouldechova (2017)).

One main attraction of statistical definitions of fairness is that they can in principle be obtained and checked without making any assumptions about the underlying population, and hence lead to more immediately actionable algorithmic approaches. On the other hand, individual notions of fairness ask for the algorithm to satisfy some guarantee which binds at the individual, rather than group, level. This often has the semantics that "individuals who are similar" should be treated "similarly", or "less qualified individuals should not be favored over more qualified individuals" (Joseph et al., 2016).

Interestingly, one of the earliest notions of fairness proposed in machine learning observes that their definition is in fact a generalization of the notion of differential privacy Dwork et al. (2012):

Definition 1.3.2 (Dwork et al. (2012))**.** *Fixing a distance metric d between individuals, and a distance metric between distributions over outcomes D, a classifier $\mathcal{A}$ is fair if for every x, y we have:*

$$D(\mathcal{A}x, \mathcal{A}y) \leq d(x,y)$$

Individual notions of fairness have attractively strong semantics, but their main drawback is that achieving them seemingly requires more assumptions to be made about the setting under consideration.

The fairness chapters in this thesis focus on mitigating the practical drawbacks of these two most common types of fairness definitions: designing statistical notions that give finer grained guarantees, and solving the problem of choosing a metric d for individual notions of fairness by learning d from data gathered from a panel of experts. In the next section we elaborate more on the results obtained.

1.4. Results

Broadly speaking the first two chapters focus on removing existing practical obstacles to differentially private machine learning, and are in the spirit of other papers completed during the Ph.D. but not included in the thesis [Joseph et al. (2019), Neel et al. (2019)]. Chapters 5, 6 focus on mitigating the shortcomings of popular notions of fairness in supervised learning, and developing learning algorithms that are fair with respect to these new fairness notions.

Chapter 3 starts with the observation that traditional approaches to differential privacy assume a fixed privacy requirement ε for a computation, and attempt to maximize the accuracy of the computation subject to the privacy constraint. As differential privacy is increasingly deployed in practical settings, it may often be that there is instead a fixed accuracy requirement for a given computation and the data analyst would like to maximize the privacy of the computation subject to the accuracy constraint. This raises the question of how to find and run a maximally private empirical risk minimizer subject to a given accuracy requirement.

- We propose a general "noise reduction" framework that can apply to a variety of private empirical risk minimization (ERM) algorithms, using them to "search" the space of privacy levels to find the empirically strongest one that meets the accuracy

constraint, incurring only logarithmic overhead in the number of privacy levels searched.

- The privacy analysis of our algorithm leads naturally to a new version of differential privacy where the privacy parameters are dependent on the data, which we term *ex-post* privacy, and which is related to the recently introduced notion of privacy odometers.

- We also give an *ex-post* privacy analysis of the classical AboveThreshold privacy tool, modifying it to allow for queries chosen depending on the database.

- Finally, we apply our approach to two common objectives, regularized linear and logistic regression, and empirically compare our noise reduction methods to (i) inverting the theoretical utility guarantees of standard private ERM algorithms and (ii) a stronger, empirical baseline based on binary search.

Chapter 4 develops theory for using *heuristics* to solve computationally hard problems in differential privacy. Heuristic approaches have enjoyed tremendous success in machine learning, for which performance can be empirically evaluated. However, privacy guarantees cannot be evaluated empirically, and must be proven — without making heuristic assumptions.

- We show that learning problems over broad classes of functions — those that have polynomially sized universal identification sets — can be solved privately and efficiently, assuming the existence of a non-private oracle for solving the same problem.

- Our first algorithm yields a privacy guarantee that is contingent on the correctness of the oracle. We then give a reduction which applies to a class of heuristics which we call *certifiable*, which allows us to convert oracle-dependent privacy guarantees to worst-case privacy guarantee that hold even when the heuristic standing in for the oracle might fail in adversarial ways.

Finally, we consider classes of functions for which both they and their dual classes have small universal identification sets. This includes most classes of simple boolean functions studied in the PAC learning literature, including conjunctions, disjunctions, parities, and discrete halfspaces.

- We show that there is an efficient algorithm for privately constructing synthetic data for any such class, given a non-private learning oracle. 'This in particular gives the first oracle-efficient algorithm for privately generating synthetic data for contingency tables.

The most intriguing question left open by our work is whether or not *every problem* that can be solved differentially privately can be privately solved with an oracle-efficient algorithm. While we do not resolve this, we give a barrier result that suggests that any generic oracle-efficient reduction must fall outside of a natural class of algorithms (which includes the algorithms given in this chapter).

Chapter 5 aims to solve the so-called *gerrymandering* problem with statistical fairness notions. This is the situation in which a classifier appears to be fair on each individual group, but badly violates the fairness constraint on one or more structured *subgroups* defined over the protected attributes (such as certain combinations of protected attribute values), which we call rich subgroup fairness. We propose instead to demand statistical notions of fairness across exponentially (or infinitely) many subgroups, defined by a structured class of functions over the protected attributes. This interpolates between statistical definitions of fairness, and recently proposed individual notions of fairness, but it raises several computational challenges. It is no longer clear how to even check or *audit* a fixed classifier to see if it satisfies such a strong definition of fairness. Our results address both the problem of auditing with respect to rich subgroup fairness, and learning the optimally accurate rich subgroup fair classifier. We show:

- The computational problem of auditing subgroup fairness for both equality of false positive rates and statistical parity is equivalent to the problem of weak agnostic learning.

This means it is computationally hard in the worst case, even for simple structured subclasses. However, it also suggests that common heuristics for learning can be applied to successfully solve the auditing problem in practice.

- We then derive two algorithms that provably converge to the best fair distribution over classifiers in a given class, given access to oracles which can optimally solve the agnostic learning problem.

The algorithms are based on a formulation of subgroup fairness as a two-player zero-sum game between a Learner (the primal player) and an Auditor (the dual player). Both algorithms compute an equilibrium of this game. We obtain our first algorithm by simulating play of the game by having Learner play an instance of the no-regret *Follow the Perturbed Leader* algorithm, and having Auditor play best response. This algorithm provably converges to an approximate Nash equilibrium (and thus to an approximately optimal subgroup-fair distribution over classifiers) in a polynomial number of steps. We obtain our second algorithm by simulating play of the game by having both players play *Fictitious Play*, which enjoys only provably asymptotic convergence, but has the merit of simplicity and faster per-step computation. We implement the Fictitious Play version using linear regression as a heuristic oracle, and show that we can effectively both audit and learn fair classifiers on real datasets.

Chapter 6 solves the problem of implementing individual fairness without knowing the distance metric, under the assumption that we have access to random pairs from the data distribution scored by the distance metric. We make no assumption that these pairs are consistent with any metric. By formulating the problem of learning the optimal fair

classifier as computing an approximate Nash equilibrium of a two-player zero-sum game, using techniques similar to those developed in Chapter 5 we:

- Provide a provably convergent oracle-efficient algorithm for minimizing error subject to the fairness constraints, and prove generalization theorems for both accuracy and fairness.

Since the constrained pairs could be elicited either from a panel of judges, or from particular individuals, our framework also provides a means for algorithmically enforcing subjective notions of fairness.

- We report on preliminary findings of a behavioral study of subjective fairness using human-subject fairness constraints elicited on the COMPAS criminal recidivism dataset.

CHAPTER 2

Background

We now develop key definitions and statements that are used throughout this thesis. We first formally define differential privacy, and some of its basic properties, which appear in Chapters 3, 4. We then define the types of optimization oracles and notions of oracle-efficiency and dependence that are used in Chapters 4, 5, 6.

2.1. Differential Privacy

Let $\mathcal{X}$ denote a d-dimensional data domain (e.g. $\mathbb{R}^d$ or $\{0,1\}^d$). We write n to denote the size of a dataset S. We call two *data sets* $S, S' \in \mathcal{X}^n$ *neighbors* (written as $S \sim S'$) if S can be derived from S' by replacing a single data point with some other element of $\mathcal{X}$. We can now define the notion of approximate or (ε, δ)-differential privacy.

Definition 2.1.1 (Differential Privacy Dwork et al. (2006b,c))**.** *Fix $\varepsilon, \delta \geq 0$. A randomized algorithm $\mathcal{A} : \mathcal{X}^* \to \mathcal{O}$ is (ε, δ)-differentially private if for every pair of neighboring data sets $S \sim S' \in \mathcal{X}^*$, and for every event $\Omega \subseteq \mathcal{O}$:*

$$\Pr[\mathcal{A}(S) \in \Omega] \leq \exp(\varepsilon) \Pr[\mathcal{A}(S') \in \Omega] + \delta.$$

We say $\mathcal{A}$ is ε-differentially private or satisfies ε-pure differential privacy if $\delta = 0$ above.

Differentially private computations enjoy two nice properties:

Theorem 2.1.2 (Post Processing Dwork et al. (2006b,c))**.** *Let $A : \mathcal{X}^* \to \mathcal{O}$ be any (ε, δ)-differentially private algorithm, and let $f : \mathcal{O} \to \mathcal{O}'$ be any function. Then the algorithm $f \circ A : \mathcal{X}^* \to \mathcal{O}'$ is also (ε, δ)-differentially private.*

Post-processing implies that, for example, every *decision* process based on the output of a differentially private algorithm is also differentially private.

Theorem 2.1.3 (Basic Composition Dwork et al. (2006b,c)). *Let $\mathcal{A}_1 : \mathcal{X}^* \to \mathcal{O}$, $\mathcal{A}_2 : \mathcal{O} \times \mathcal{X}^* \to \mathcal{O}'$ be such that $\mathcal{A}_1$ is $(\varepsilon_1, \delta_1)$-differentially private, and $\mathcal{A}_2(o, \cdot)$ is $(\varepsilon_2, \delta_2)$-differentially private for every $o \in \mathcal{O}$. Then the algorithm $A : \mathcal{X}^* \to \mathcal{O}'$ defined as $A(x) = \mathcal{A}_2(\mathcal{A}_1(x), x)$ is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-differentially private.*

The Laplace distribution plays a fundamental role in differential privacy. The Laplace Distribution centered at 0 with scale b is the distribution with probability density function $\mathrm{Lap}(z|b) = \frac{1}{2b} e^{-\frac{|z|}{b}}$. We write $X \sim \mathrm{Lap}(b)$ when X is a random variable drawn from a Laplace distribution with scale b. Let $f : \mathcal{X}^n \to \mathbb{R}^k$ be an arbitrary function. The ℓ_1 *sensitivity* of f is defined to be $\Delta_1(f) = \max_{S \sim S'} \|f(S) - f(S')\|_1$. The *Laplace mechanism* with parameter ε simply adds noise drawn independently from $\mathrm{Lap}\left(\frac{\Delta_1(f)}{\varepsilon}\right)$ to each coordinate of $f(S)$.

Theorem 2.1.4 ((Dwork et al., 2006b)). *The Laplace mechanism is ε-differentially private.*

2.2. Oracle-Efficient Algorithms

In chapters 4, 5, and 6 will often refer to optimization oracles and oracle-efficient algorithms. It will be useful for us to study oracles that solve weighted generalizations of the minimization problem, in which each datapoint $x_i \in S$ is paired with a real-valued weight w_i. In the literature on oracle-efficiency in machine learning, these are widely employed, and are known as *cost-sensitive classification oracles*. Via a simple translation and re-weighting argument, they are no more powerful than agnostic learning oracles Zadrozny et al. (2003), but are more convenient to work with. We formally define weak agnostic learning (Chapter 5), weighted (private) optimization oracles (Chapter 4), and cost-sensitive classification below (Chapters 5, 6). We will also define the notion of *certifiability* of an oracle (Chapter 4), which is an oracle that may not perfectly solve the optimization problem it is given, but when it does fail outputs "FAIL". Throughout the chapters we will use statistical queries $\mathcal{Q}$ interchangeably with a hypothesis class $\mathcal{H}$. Specifically we refer to queries in the context of private data release, and hypothesis classes in the context of learning; in either case they are simply functions from the data domain $\mathcal{X}$ to $[0, 1]$ or $\{0, 1\}$ depending on the context.

Definition 2.2.1 (Weak Agnostic Learning (Kearns et al., 1994; Kalai et al., 2008))**.** *Let Q be a distribution over $\mathcal{X} \times \{0,1\}$ and let $\varepsilon, \varepsilon' \in (0, 1/2)$ such that $\varepsilon \geq \varepsilon'$. We say that the hypothesis class $\mathcal{H}$ is* $(\varepsilon, \varepsilon')$-weakly agnostically learnable *under distribution Q if there exists an algorithm L such that when given sample access to Q, L runs in time* $\text{poly}(1/\varepsilon', 1/\delta)$*, and with probability $1 - \delta$, outputs a hypothesis $h \in \mathcal{H}$ such that*

$$\min_{f \in \mathcal{H}} err(f, Q) \leq 1/2 - \varepsilon \implies err(h, Q) \leq 1/2 - \varepsilon'.$$

where $err(h, Q) = \Pr_{(x,y)\sim Q}[h(x) \neq y]$.

Formally, an instance of a cost-sensitive classification (CSC) problem for the class $\mathcal{H}$ is given by a set of n tuples $\{(X_i, c_i^0, c_i^1)\}_{i=1}^n$ such that c_i^ℓ corresponds to the cost for predicting label ℓ on point X_i. Given such an instance as input, a CSC oracle finds a hypothesis $\hat{h} \in \mathcal{H}$ that minimizes the total cost across all points:

$$\hat{h} \in \operatorname*{argmin}_{h \in \mathcal{H}} \sum_{i=1}^{n} [h(X_i) c_i^1 + (1 - h(X_i)) c_i^0] \tag{2.1}$$

A crucial property of a CSC problem is that the solution is invariant to translations of the costs.

Claim 2.2.2. *Let $\{(X_i, c_i^0, c_i^1)\}_{i=1}^n$ be a CSC instance, and $\{(\tilde{c}_i^0, \tilde{c}_i^1)\}$ be a set of new costs such that there exist $a_1, a_2, \ldots, a_n \in \mathbb{R}$ such that $\tilde{c}_i^\ell = c_i^\ell + a_i$ for all i and ℓ. Then*

$$\operatorname*{argmin}_{h \in \mathcal{H}} \sum_{i=1}^{n} [h(X_i) c_i^1 + (1 - h(X_i)) c_i^0] = \operatorname*{argmin}_{h \in \mathcal{H}} \sum_{i=1}^{n} [h(X_i) \tilde{c}_i^1 + (1 - h(X_i)) \tilde{c}_i^0]$$

We now define certifiability, and the notion of an oracle-dependent algorithm.

Definition 2.2.3. *A weighted optimization oracle for a class of statistical queries $\mathcal{Q}$ is a function $\mathcal{O}^* : (\mathcal{X} \times \mathbb{R})^* \to \mathcal{Q}$ takes as input a weighted dataset $WD \in (\mathcal{X} \times \mathbb{R})^*$ and outputs a query $q = \mathcal{O}^*(WD)$ such that*

$$q \in \operatorname*{argmin}_{q^* \in \mathcal{Q}} \sum_{(x_i, w_i) \in WD} w_i q^*(x_i).$$

Definition 2.2.4. *A certifiable heuristic optimization oracle for a class of queries $\mathcal{Q}$ is a polynomial time algorithm $\mathcal{O} : (\mathcal{X} \times \mathbb{R})^* \to (\mathcal{Q} \cup \perp)$ that takes as input a weighted dataset $WD \in (\mathcal{X} \times \mathbb{R})^*$ and either outputs $\mathcal{O}(WD) = q \in \text{argmin}_{q^* \in \mathcal{Q}} \sum_{(x_i, w_i) \in WD} w_i q^*(x_i)$ or else outputs $\perp$ ("Fail"). If it outputs a statistical query q, we say the oracle has succeeded.*

In contrast, a heuristic optimization oracle (that is not certifiable) has no guarantees of correctness. Without loss of generality, such oracles never need to return "Fail" (since they can always instead output a default statistical query in this case).

Definition 2.2.5. *A (non-certifiable) heuristic optimization oracle for a class of queries $\mathcal{Q}$ is an arbitrary polynomial time algorithm $M : (\mathcal{X} \times \mathbb{R})^* \to \mathcal{Q}$. Given a call to the oracle defined by a weighted dataset $WD \in (\mathcal{X} \times \mathbb{R})^*$ we say that the oracle has succeeded on this call up to error α if it outputs a query q such that $\sum_{(x_i, w_i) \in WD} w_i q(x_i) \leq \min_{q^* \in \mathcal{Q}} \sum_{(x_i, w_i) \in WD} w_i q^*(x_i) + \alpha$. If it succeeds up to error 0, we just say that the heuristic oracle has succeeded. Note that there may not be any efficient procedure to determine whether the oracle has succeeded up to error α.*

We say an algorithm $\mathcal{A}_{\mathcal{O}}$ is (certifiable)-oracle dependent if throughout the course of its run it makes a series of (possibly adaptive) calls to a (certifiable) heuristic optimization oracle $\mathcal{O}$. An oracle-dependent algorithm $\mathcal{A}_{\mathcal{O}}$ is *oracle equivalent* to an algorithm $\mathcal{A}$ if given access to a perfect optimization oracle $\mathcal{O}^*$, $\mathcal{A}_{\mathcal{O}^*}$ induces the same distribution on outputs as $\mathcal{A}$. We now state an intuitive lemma (that could also be taken as a more formal definition of *oracle equivalence*). See the Appendix for a proof.

Lemma 2.2.6. *Let $\mathcal{A}_{\mathcal{O}}$ be a certifiable-oracle dependent algorithm that is oracle equivalent to $\mathcal{A}$. Then for any fixed input dataset S, there exists a coupling between $\mathcal{A}(S)$ and $\mathcal{A}_{\mathcal{O}}(S)$ such that* $\Pr[\mathcal{A}_{\mathcal{O}}(S) = a | \mathcal{A}_{\mathcal{O}}(S) \neq \perp] = \Pr[\mathcal{A}(S) = a | \mathcal{A}_{\mathcal{O}}(S) \neq \perp]$.

We will also discuss differentially private heuristic optimization oracles, in order to state additional consequences of our construction in Section 4.4. Note that because differential privacy precludes exact computations, differentially private heuristic oracles are necessarily non-certifiable, and will never succeed up to error 0.

Definition 2.2.7. *A weighted (ε, δ)-differentially private (α, β)-accurate learning oracle for a class of statistical queries $\mathcal{Q}$ is an (ε, δ) differentially private algorithm $\mathcal{O} : (\mathcal{X} \times \mathbb{R})^* \to C$ that takes as input a weighted dataset $WD \in (\mathcal{X} \times \mathbb{R})^*$ and outputs a query $q_{priv} \in \mathcal{Q}$ such that with probability $1 - \beta$:*

$$\sum_{(x_i, w_i) \in WD} w_i q_{priv}(x_i) - \operatorname*{argmin}_{q^* \in C} \sum_{(x_i, w_i) \in WD} w_i q^*(x_i) \leq \alpha$$

We say that an algorithm is *oracle-efficient* if given access to an oracle (in this chapter, always a weighted optimization oracle for a class of statistical queries) it runs in polynomial time in the length of its input, and makes a polynomial number of calls to the oracle. In practice, we will be interested in the performance of oracle-efficient algorithms when they are instantiated with heuristic oracles. Thus, we further require oracle-efficient algorithms to halt in polynomial time even when the oracle fails. When we design algorithms for optimization and synthetic data generation problems, their (α, β)-accuracy guarantees will generally rely on all queries to the oracle succeeding (possibly up to error $O(\alpha)$).

CHAPTER 3

Accuracy First: Selecting a Differential Privacy in ERM

3.1. Introduction

While differential privacy enjoys over a decade of study as a theoretical construct, practical deployments, including by Google (Fanti et al., 2015) and Apple (Greenberg, 2016), are a much more recent occurrence. As the large theoretical literature is put into practice, we start to see disconnects between assumptions implicit in the theory and the practical necessities of applications. In this chapter we focus our attention on one such assumption in the domain of private empirical risk minimization (ERM): that the data analyst first chooses a privacy requirement, and then attempts to obtain the best accuracy guarantee (or empirical performance) that she can, given the chosen privacy constraint. Existing theory is tailored to this view: the data analyst can pick her privacy parameter ε via some exogenous process, and either plug it into a "utility theorem" to upper bound her accuracy loss, or simply deploy her algorithm and (privately) evaluate its performance.

While existing algorithms take a privacy-first perspective, in practice, product requirements may impose hard accuracy constraints, and privacy (while desirable) may not be the overriding concern. In such situations, things are reversed: the data analyst first fixes an accuracy requirement, and then would like to find the smallest privacy parameter consistent with the accuracy constraint. Here, we find a gap between theory and practice. The only theoretically sound method available is to take a "utility theorem" for an existing private ERM algorithm and solve for the smallest value of ε (the differential privacy parameter)—and other parameter values that need to be set—consistent with her accuracy requirement, and then run the private ERM algorithm with the resulting ε. But because utility theorems tend to be worst-case bounds, this approach will generally be extremely conservative, leading to a much larger value of ε (and hence a much larger leakage of information) than is necessary for the problem at hand. Alternately, the analyst could attempt an empirical

search for the smallest value of ε consistent with her accuracy goals. However, because this search is itself a data-dependent computation, it incurs the overhead of additional privacy loss. Furthermore, it is not *a priori* clear how to undertake such a search with nontrivial privacy guarantees for two reasons: first, the worst case could involve a very long search which reveals a large amount of information, and second, the selected privacy parameter is now itself a data-dependent quantity, and so it is not sensible to claim a "standard" guarantee of differential privacy for any finite value of ε ex-ante.

In this chapter, we describe, analyze, and empirically evaluate a principled variant of this second approach, which attempts to empirically find the smallest value of ε consistent with an accuracy requirement. We give a meta-method that can be applied to several interesting classes of private learning algorithms and introduces very little privacy overhead as a result of the privacy-parameter search. Conceptually, our meta-method initially computes a very private hypothesis, and then gradually subtracts noise (making the computation less and less private) until a sufficient level of accuracy is achieved. One key technique that saves significant factors in privacy loss over naive search is the use of correlated noise generated by the method of (Koufogiannis et al., 2017), which formalizes the conceptual idea of "subtracting" noise without incurring additional privacy overhead. In order to select the most private of these queries that meets the accuracy requirement, we introduce a natural modification of the now-classic AboveThreshold algorithm (Dwork and Roth, 2014a), which iteratively checks a sequence of queries on a dataset and privately releases the index of the first to approximately exceed some fixed threshold. Its privacy cost increases only logarithmically with the number of queries. We provide an analysis of AboveThreshold that holds even if the queries themselves are the result of differentially private computations, showing that if AboveThreshold terminates after t queries, one only pays the privacy costs of AboveThreshold plus the privacy cost of revealing those first t private queries. When combined with the above-mentioned correlated noise technique of (Koufogiannis et al., 2017), this gives an algorithm whose privacy loss is *equal* to that of the final hypothesis output – the previous ones coming "for free" – plus the privacy loss of AboveThreshold.

Because the privacy guarantees achieved by this approach are not fixed a priori, but rather are a function of the data, we introduce and apply a new, corresponding privacy notion, which we term *ex-post* privacy, and which is closely related to the recently introduced notion of "privacy odometers" (Rogers et al., 2016).

In Section 3.4, we empirically evaluate our noise reduction meta-method, which applies to any ERM technique which can be described as a post-processing of the Laplace mechanism. This includes both direct applications of the Laplace mechanism, like *output perturbation* (Chaudhuri et al., 2011); and more sophisticated methods like *covariance perturbation* (Smith et al., 2017), which perturbs the covariance matrix of the data and then performs an optimization using the noisy data. Our experiments concentrate on ℓ_2 regularized least-squares regression and ℓ_2 regularized logistic regression, and we apply our noise reduction meta-method to both output perturbation and covariance perturbation. Our empirical results show that the active, ex-post privacy approach massively outperforms inverting the theory curve, and also improves on a baseline "ε-doubling" approach.

3.2. Ex-post Differential Privacy

3.2.1. Definition

It is possible to design computations that do not satisfy the differential privacy definition, but whose outputs are private to an extent that can be quantified after the computation halts. For example, consider an experiment that repeatedly runs an ε'-differentially private algorithm, until a stopping condition defined by the output of the algorithm itself is met. This experiment does not satisfy ε-differential privacy any fixed value of ε, since there is no fixed maximum number of rounds for which the experiment will run (for a fixed number of rounds, a simple composition theorem, 2.1.3, shows that the ε-guarantees in a sequence of computations "add up.") However, if ex-post we see that the experiment has stopped after k rounds, the data can in some sense be assured an "ex-post privacy loss" of only $k\varepsilon'$. Rogers et al. Rogers et al. (2016) initiated the study of *privacy odometers*, which formalize this idea.

Conceptually, if a data scientist is constrained to obey a differential privacy constraint of ε over all computations run on a particular data set, then she can make better use of her data by accounting for the actual, ex post, privacy losses of her algorithms rather than their worst-case upper bounds.

We apply a related idea here, for a different purpose. Our goal is to design algorithms that always achieve a target accuracy but that may have variable privacy levels depending on their input.

Definition 3.2.1. *Given a randomized algorithm $\mathcal{A} : \mathcal{X}^* \to \mathcal{O}$, define the* ex-post privacy loss[1] *of $\mathcal{A}$ on outcome o to be*

$$Loss(o) = \max_{D,D':D\sim D'} \log \frac{\Pr[\mathcal{A}(D) = o]}{\Pr[\mathcal{A}(D') = o]}.$$

We refer to $\exp(\text{Loss}(o))$ as the *ex-post privacy risk* factor.

Definition 3.2.2 (Ex-Post Differential Privacy)**.** *Let $\mathcal{E} : \mathcal{O} \to (\mathbb{R}_{\geq 0} \cup \{\infty\})$ be a function on the outcome space of algorithm $\mathcal{A} : \mathcal{X}^* \to \mathcal{O}$. Given an outcome $o = A(D)$, We say that $\mathcal{A}$ satisfies $\mathcal{E}(o)$-ex-post differential privacy if for all $o \in \mathcal{O}$, $Loss(o) \leq \mathcal{E}(o)$.*

Note that if $\mathcal{E}(o) \leq \varepsilon$ for all o, $\mathcal{A}$ is ε-differentially private. Ex-post differential privacy has the same semantics as differential privacy, once the output of the mechanism is known: it bounds the log-likelihood ratio of the dataset being D vs. D', which controls how an adversary with an arbitrary prior on the two cases can update her posterior.

3.2.2. Gradual Private Release

Koufogiannis et al. Koufogiannis et al. (2017) study how to gradually release private data using the Laplace mechanism with an increasing sequence of ε values, with a privacy cost scaling only with the privacy of the *marginal* distribution on the least private release, rather than the sum of the privacy costs of independent releases. For intuition, the algorithm can be pictured as a continuous random walk starting at some private data v with the property

[1] If $\mathcal{A}$'s output is from a continuous distribution rather than discrete, we abuse notation and write $\Pr[\mathcal{A}(D) = o]$ to mean the probability density at output o.

that the marginal distribution at each point in time is Laplace centered at v, with variance increasing over time. Releasing the value of the random walk at a fixed point in time gives a certain output distribution, for example, $\hat{v}$, with a certain privacy guarantee ε. To produce $\hat{v}'$ whose *ex-ante* distribution has higher variance (is more private), one can simply "fast forward" the random walk from a starting point of $\hat{v}$ to reach $\hat{v}'$; to produce a less private $\hat{v}'$, one can "rewind." The total privacy cost is $\max\{\varepsilon, \varepsilon'\}$ because, given the "least private" point (say $\hat{v}$), all "more private" points can be derived as post-processings given by taking a random walk of a certain length starting at $\hat{v}$. Note that were the Laplace random variables used for each release independent, the composition theorem would require *summing* the ε values of all releases.

In our private algorithms, we will use this noise reduction mechanism as a building block to generate a list of private hypotheses $\theta^1, \ldots, \theta^T$ with gradually increasing ε values. Importantly, releasing any prefix $(\theta^1, \ldots, \theta^t)$ only incurs the privacy loss in θ^t. More formally:

Algorithm 1 Noise Reduction Koufogiannis et al. (2017): $\mathrm{NR}(v, \Delta, \{\varepsilon_t\})$

Input: private vector v, sensitivity parameter Δ, list $\varepsilon_1 < \varepsilon_2 < \cdots < \varepsilon_T$
Set $\hat{v}_T := v + \mathrm{Lap}(\Delta/\varepsilon_T)$ ▷ drawn i.i.d. for each coordinate
for $t = T-1, T-2, \ldots, 1$ **do**
  With probability $\left(\frac{\varepsilon_t}{\varepsilon_{t+1}}\right)^2$: set $\hat{v}_t := \hat{v}_{t+1}$
  Else: set $\hat{v}_t := \hat{v}_{t+1} + \mathrm{Lap}(\Delta/\varepsilon_t)$ ▷ drawn i.i.d. for each coordinate
Return $\hat{v}_1, \ldots, \hat{v}_T$

Theorem 3.2.3 (Koufogiannis et al. (2017))**.** *Let f have ℓ_1 sensitivity Δ and let $\hat{v}_1, \ldots, \hat{v}_T$ be the output of Algorithm 1 on $v = f(D)$, Δ, and the increasing list $\varepsilon_1, \ldots, \varepsilon_T$. Then for any t, the algorithm which outputs the prefix $(\hat{v}_1, \ldots, \hat{v}_t)$ is ε_t-differentially private.*

3.2.3. AboveThreshold with Private Queries

Our high-level approach to our eventual ERM problem will be as follows: Generate a sequence of hypotheses $\theta_1, \ldots, \theta_T$, each with increasing accuracy and decreasing privacy; then test their accuracy levels sequentially, outputting the first one whose accuracy is "good

enough." The classical AboveThreshold algorithm (Dwork and Roth, 2014a) takes in a dataset and a sequence of queries and privately outputs the index of the first query to exceed a given threshold (with some error due to noise). We would like to use AboveThreshold to perform these accuracy checks, but there is an important obstacle: for us, the "queries" themselves depend on the private data. [2] A standard composition analysis would involve first privately publishing *all* the queries, then running AboveThreshold on these queries (which are now public). Intuitively, though, it would be much better to generate and publish the queries one at a time, until AboveThreshold halts, at which point one would not publish any more queries. The problem with analyzing this approach is that, a-priori, we do not know when AboveThreshold will terminate; to address this, we analyze its *ex-post privacy*.[3]

Algorithm 2 InteractiveAboveThreshold: IAT$(D, \varepsilon, W, \Delta, M)$

Input: Dataset D, privacy loss ε, threshold W, ℓ_1 sensitivity Δ, algorithm M
Let $\hat{W} = W + \mathrm{Lap}\left(\frac{2\Delta}{\varepsilon}\right)$
for each query $t = 1, \ldots, T$ **do**
  Query $f_t \leftarrow M(D)_t$
  if $f_t(D) + \mathrm{Lap}\left(\frac{4\Delta}{\varepsilon}\right) \geq \hat{W}$: **then** Output (t, f_t); **Halt.**
Output $(T, \perp)$.

Let us say that an algorithm $M(D) = (f_1, \ldots, f_T)$ is $(\varepsilon_1, \ldots, \varepsilon_T)$-*prefix-private* if for each t, the function that runs $M(D)$ and outputs just the prefix $(f_1, \ldots, f_t)$ is ε_t-differentially private.

Lemma 3.2.4. *Let $M : \mathcal{X}^* \to (\mathcal{X}^* \to \mathcal{O})^T$ be a $(\varepsilon_1, \ldots, \varepsilon_T)$-prefix private algorithm that returns T queries, and let each query output by M have ℓ_1 sensitivity at most Δ. Then Algorithm 2 run on D, ε_A, W, Δ, and M is $\mathcal{E}$-ex-post differentially private for $\mathcal{E}((t, \cdot)) = \varepsilon_A + \varepsilon_t$ for any $t \in [T]$.*

The proof, which is a variant on the proof of privacy for AboveThreshold Dwork and Roth (2014a), appears in the Appendix, along with an accuracy theorem for IAT.

[2] In fact, there are many applications beyond our own in which the sequence of queries input to AboveThreshold might be the result of some private prior computation on the data, and where we would like to release both the stopping index of AboveThreshold and the "query object." (In our case, the query objects will be parameterized by learned hypotheses $\theta_1, \ldots, \theta_T$.)

[3] This result does not follow from a straightforward application of privacy odometers from Rogers et al. (2016), because the privacy analysis of algorithms like the noise reduction technique is not compositional.

Remark 3.2.5. *Throughout we study ε-differential privacy, instead of the weaker (ε, δ) (approximate) differential privacy. Part of the reason is that an analogue of 3.2.4 does not seem to hold for (ε, δ)-differentially private queries without further assumptions, as the necessity to union-bound over the δ "failure probability" that the privacy loss is bounded for each query can erase the ex-post gains. We leave obtaining similar results for approximate differential privacy as an open problem.*

3.3. Noise-Reduction with Private ERM

In this section, we provide a general private ERM framework that allows us to approach the best privacy guarantee achievable on the data given a target excess risk goal. Throughout the section, we consider an input dataset D that consists of n row vectors $X_1, X_2, \ldots, X_n \in \mathbb{R}^p$ and a column $y \in \mathbb{R}^n$. We will assume that each $\|X_i\|_1 \leq 1$ and $|y_i| \leq 1$. Let $d_i = (X_i, y_i) \in \mathbb{R}^{p+1}$ be the i-th data record. Let ℓ be a loss function such that for any hypothesis θ and any data point (X_i, y_i) the loss is $\ell(\theta, (X_i, y_i))$. Given an input dataset D and a regularization parameter λ, the goal is to minimize the following regularized empirical loss function over some feasible set C:

$$L(\theta, D) = \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, (X_i, y_i)) + \frac{\lambda}{2} \|\theta\|_2^2.$$

Let $\theta^* = \operatorname{argmin}_{\theta \in C} \ell(\theta, D)$. Given a target accuracy parameter α, we wish to privately compute a θ_p that satisfies $L(\theta_p, D) \leq L(\theta^*, D) + \alpha$, while achieving the best ex-post privacy guarantee. For simplicity, we will sometimes write $L(\theta)$ for $L(\theta, D)$.

One simple baseline approach is a "doubling method": Start with a small ε value, run an ε-differentially private algorithm to compute a hypothesis θ and use the Laplace mechanism to estimate the excess risk of θ; if the excess risk is lower than the target, output θ; otherwise double the value of ε and repeat the same process. (See the appendix for details.) As a result, we pay for privacy loss for every hypothesis we compute and every excess risk we estimate.

Our meta-method provides a more cost-effective way to select the privacy level. The algorithm takes a more refined set of privacy levels $\varepsilon_1 < \ldots < \varepsilon_T$ as input and generates a sequence of hypotheses $\theta^1, \ldots, \theta^T$ such that the generation of each θ^t is ε_t-private. Then it releases the hypotheses θ^t in order, halting when a hypothesis meets the accuracy goal. Importantly, there are two key components that reduce the privacy loss in our method:

1. We use 1, the "noise reduction" method of Koufogiannis et al. (2017), for generating the sequence of hypotheses: we first compute a very private and noisy θ^1, and then obtain the subsequent hypotheses by gradually "de-noising" θ^1. As a result, any prefix $(\theta^1, \ldots, \theta^k)$ incurs a privacy loss of only ε_k (as opposed to $(\varepsilon_1 + \ldots + \varepsilon_k)$ if the hypotheses were independent).

2. When evaluating the excess risk of each hypothesis, we use 2, InteractiveAboveThreshold, to determine if its excess risk exceeds the target threshold. This incurs substantially less privacy loss than independently evaluating the excess risk of each hypothesis using the Laplace mechanism (and hence allows us to search a finer grid of values).

For the rest of this section, we will instantiate our method concretely for two ERM problems: ridge regression and logistic regression. In particular, our noise-reduction method is based on two private ERM algorithms: the recently introduced covariance perturbation technique of Smith et al. (2017), and output perturbation Chaudhuri et al. (2011).

3.3.1. Covariance Perturbation for Ridge Regression

In ridge regression, we consider the squared loss function: $\ell((X_i, y_i), \theta) = \frac{1}{2}(y_i - \langle \theta, X_i \rangle)^2$, and hence empirical loss over the data set is defined as

$$L(\theta, D) = \frac{1}{2n}\|y - X\theta\|_2^2 + \frac{\lambda\|\theta\|_2^2}{2},$$

where X denotes the $(n \times p)$ matrix with row vectors $X_1, \ldots, X_n$ and $y = (y_1, \ldots, y_n)$. Since the optimal solution for the unconstrained problem has ℓ_2 norm no more than $\sqrt{1/\lambda}$

(see the appendix for a proof), we will focus on optimizing θ over the constrained set $C = \{a \in \mathbb{R}^p \mid \|a\|_2 \leq \sqrt{1/\lambda}\}$, which will be useful for bounding the ℓ_1 sensitivity of the empirical loss.

Before we formally introduce the covariance perturbation algorithm due to Smith et al. (2017), observe that the optimal solution θ^* can be computed as

$$\theta^* = \operatorname*{argmin}_{\theta \in C} L(\theta, D) = \operatorname*{argmin}_{\theta \in C} \frac{(\theta^\intercal (X^\intercal X)\theta - 2\langle X^\intercal y, \theta \rangle)}{2n} + \frac{\lambda \|\theta\|_2^2}{2}.$$

In other words, θ^* only depends on the private data through $X^\intercal y$ and $X^\intercal X$. To compute a private hypothesis, the covariance perturbation method simply adds Laplace noise to each entry of $X^\intercal y$ and $X^\intercal X$ (the covariance matrix), and solves the optimization based on the noisy matrix and vector. The formal description of the algorithm and its guarantee are in 3.3.1. Our analysis slightly deviates from the one in Smith et al. (2017) because that paper considers the "local privacy" setting, and also adds Gaussian noise whereas we use Laplace. The proof is deferred to the appendix.

Theorem 3.3.1. *Fix any $\varepsilon > 0$. For any input data set D, consider the mechanism $\mathcal{M}$ that computes*

$$\theta_p = \operatorname*{argmin}_{\theta \in C} \frac{1}{2n} (\theta^\intercal (X^\intercal X + B)\theta - 2\langle X^\intercal y + b, \theta \rangle) + \frac{\lambda \|\theta\|_2^2}{2},$$

where $B \in \mathbb{R}^{p \times p}$ and $b \in \mathbb{R}^{p \times 1}$ are random Laplace matrices such that each entry of B and b is drawn from $Lap(4/\varepsilon)$. Then $\mathcal{M}$ satisfies ε-differential privacy and the output θ_p satisfies

$$\mathop{\mathbb{E}}_{B,b}\left[L(\theta_p) - L(\theta^*)\right] \leq \frac{4\sqrt{2}(2\sqrt{p/\lambda} + p/\lambda)}{n\varepsilon}.$$

In our algorithm CovNR, we will apply the noise reduction method, 1, to produce a sequence of noisy versions of the private data $(X^\intercal X, X^\intercal y)$: $(Z^1, z^1), \ldots, (Z^T, z^T)$, one for each privacy level. Then for each (Z^t, z^t), we will compute the private hypothesis by solving the

noisy version of the optimization problem in 3.1. The full description of our algorithm CovNR is in 3, and satisfies the following guarantee:

Algorithm 3 Covariance Perturbation with Noise-Reduction: CovNR$(D, \{\varepsilon_1, \ldots, \varepsilon_T\}, \alpha, \gamma)$

Input: private data set $D = (X, y)$, accuracy parameter α, privacy levels $\varepsilon_1 < \varepsilon_2 < \ldots < \varepsilon_T$, and failure probability γ

Instantiate InteractiveAboveThreshold: $\mathcal{A} = \text{IAT}(D, \varepsilon_0, -\alpha/2, \Delta, \cdot)$ with $\varepsilon_0 = 16\Delta(\log(2T/\gamma))/\alpha$ and $\Delta = (\sqrt{1/\lambda}+1)^2/(n)$

Let $C = \{a \in \mathbb{R}^p \mid \|a\|_2 \leq \sqrt{1/\lambda}\}$ and $\theta^* = \operatorname{argmin}_{\theta \in C} L(\theta)$

Compute noisy data:

$$\{Z^t\} = \text{NR}((X^\intercal X), 2, \{\varepsilon_1/2, \ldots, \varepsilon_T/2\}), \qquad \{z^t\} = \text{NR}((X^\intercal Y), 2, \{\varepsilon_1/2, \ldots, \varepsilon_T/2\})$$

for $t = 1, \ldots, T$: **do**

$$\theta^t = \operatorname*{argmin}_{\theta \in C} \frac{1}{2n}\left(\theta^\intercal Z^t \theta - 2\langle z^t, \theta\rangle\right) + \frac{\lambda\|\theta\|_2^2}{2} \tag{3.1}$$

Let $f^t(D) = L(\theta^*, D) - L(\theta^t, D)$; Query $\mathcal{A}$ with query f^t to check accuracy

if $\mathcal{A}$ returns (t, f^t) **then Output** (t, θ^t) ▷ Accurate hypothesis found.

Output: $(\perp, \theta^*)$

Theorem 3.3.2. *The instantiation of* CovNR$(D, \{\varepsilon_1, \ldots, \varepsilon_T\}, \alpha, \gamma)$ *outputs a hypothesis* θ_p *that with probability* $1 - \gamma$ *satisfies* $L(\theta_p) - L(\theta^*) \leq \alpha$. *Moreover, it is* $\mathcal{E}$*-ex-post differentially private, where the privacy loss function* $\mathcal{E}\colon ((|T| \cup \{\perp\}) \times \mathbb{R}^p) \to (\mathbb{R}_{\geq 0} \cup \{\infty\})$ *is defined as* $\mathcal{E}((k, \cdot)) = \varepsilon_0 + \varepsilon_k$ *for any* $k \neq \perp$, $\mathcal{E}((\perp, \cdot)) = \infty$, *and* $\varepsilon_0 = \frac{16(\sqrt{1/\lambda}+1)^2 \ln(2T/\gamma)}{n\alpha}$ *is the privacy loss incurred by* IAT.

3.3.2. Output Perturbation for Logistic Regression

Next, we show how to combine the output perturbation method with noise reduction for the ridge regression problem.[4] In this setting, the input data consists of n labeled examples $(X_1, y_1), \ldots, (X_n, y_n)$, such that for each i, $X_i \in \mathbb{R}^p$, $\|X_i\|_1 \leq 1$, and $y_i \in \{-1, 1\}$. The goal is to train a linear classifier given by a weight vector θ for the examples from the two classes. We consider the logistic loss function: $\ell(\theta, (X_i, y_i)) = \log(1 + \exp(-y_i \theta^\intercal X_i))$, and the empirical loss is

$$L(\theta, D) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i \theta^\intercal X_i)) + \frac{\lambda\|\theta\|_2^2}{2}.$$

[4] We study the ridge regression problem for concreteness. Our method works for any ERM problem with strongly convex loss functions.

The output perturbation method is straightforward: we simply add Laplace noise to perturb each coordinate of the optimal solution θ^*. The following is the formal guarantee of output perturbation. Our analysis deviates slightly from the one in Chaudhuri et al. (2011) since we are adding Laplace noise (see the appendix).

Theorem 3.3.3. *Fix any $\varepsilon > 0$. Let $r = \frac{2\sqrt{p}}{n\lambda\varepsilon}$. For any input dataset D, consider the mechanism that first computes $\theta^* = \operatorname{argmin}_{\theta\in\mathbb{R}^p} L(\theta)$, then outputs $\theta_p = \theta^* + b$, where b is a random vector with its entries drawn i.i.d. from $Lap(r)$. Then $\mathcal{M}$ satisfies ε-differential privacy, and θ_p has excess risk $\mathbb{E}_b\left[L(\theta_p) - L(\theta^*)\right] \leq \frac{2\sqrt{2}p}{n\lambda\varepsilon} + \frac{4p^2}{n^2\lambda\varepsilon^2}$.*

Given the output perturbation method, we can simply apply the noise reduction method NR to generate a sequence of noisy hypotheses. We will again use InteractiveAboveThreshold to check the excess risk of the hypotheses. The full algorithm OutputNR follows the same structure in 3, and we defer the formal description to the appendix.

Theorem 3.3.4. *The instantiation of* OutputNR$(D, \varepsilon_0, \{\varepsilon_1, \ldots, \varepsilon_T\}, \alpha, \gamma)$ *is $\mathcal{E}$-ex-post differentially private and outputs a hypothesis θ_p that with probability $1-\gamma$ satisfies $L(\theta_p) - L(\theta^*) \leq \alpha$, where the privacy loss function $\mathcal{E}\colon ((T]\cup\{\perp\})\times\mathbb{R}^p) \to (\mathbb{R}_{\geq 0}\cup\{\infty\})$ is defined as $\mathcal{E}((k,\cdot)) = \varepsilon_0 + \varepsilon_k$ for any $k \neq \perp$, $\mathcal{E}((\perp,\cdot)) = \infty$, and $\varepsilon_0 \leq \frac{32\ln(2T/\gamma)\sqrt{2\log 2/\lambda}}{n\alpha}$ is the privacy loss incurred by* IAT.

Proof sketch of 3.3.4. The accuracy guarantees for both algorithms follow from an accuracy guarantee of the IAT algorithm (a variant on the standard AboveThreshold bound) and the fact that we output θ^* if IAT identifies no accurate hypothesis. For the privacy guarantee, first note that any prefix of the noisy hypotheses $\theta^1, \ldots, \theta^t$ satisfies ε_t-differential privacy because of our instantiation of the Laplace mechanism (see the appendix for the ℓ_1 sensitivity analysis) and noise-reduction method NR. Then the ex-post privacy guarantee directly follows 3.2.4. □

3.4. Experiments

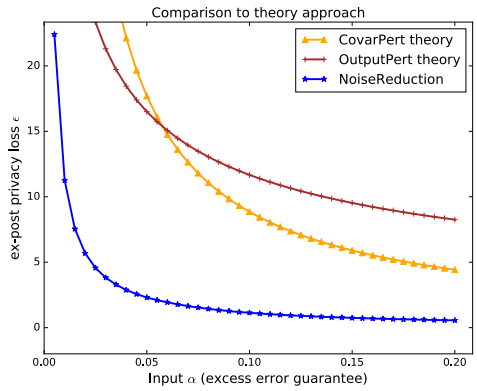
To evaluate the methods described above, we conducted empirical evaluations in two settings. We used ridge regression to predict (log) popularity of posts on Twitter in the

dataset of at Laboratoire d'Informatique de Grenoble (2017), with $p = 77$ features and subsampled to n =100,000 data points. Logistic regression was applied to classifying network events as innocent or malicious in the KDD-99 Cup dataset KDD'99 (1999), with 38 features and subsampled to 100,000 points. Details of parameters and methods appear in the appendix.

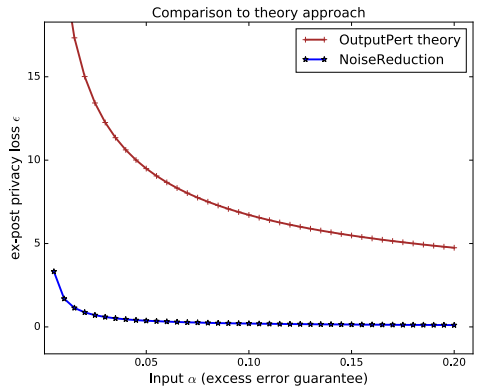
In each case, we tested the algorithm's average ex-post privacy loss for a range of input accuracy goals α, fixing a modest failure probability $\gamma = 0.1$ (and we observed that excess risks were concentrated well below $\alpha/2$, suggesting a pessimistic analysis). The results show a large improvement over the "theory" approach of simply inverting utility theorems for private ERM algorithms. (In fact, the utility theorem for the popular private stochastic gradient descent algorithm does not even give meaningful guarantees for the ranges of parameters tested; one would need an order of magnitude more data points, and even then the privacy losses are enormous, perhaps due to loose constants in the analysis.)

To gauge the more modest improvement over DoublingMethod, note that the variation in the privacy risk factor e^{ε} can still be very large; for instance, in the ridge regression setting of $\alpha = 0.05$, NoiseReduction has $e^{\varepsilon} \approx 10.0$ while Doubling has $e^{\varepsilon} \approx 495$; at $\alpha = 0.075$, the privacy risk factors are 4.65 and 56.6 respectively.

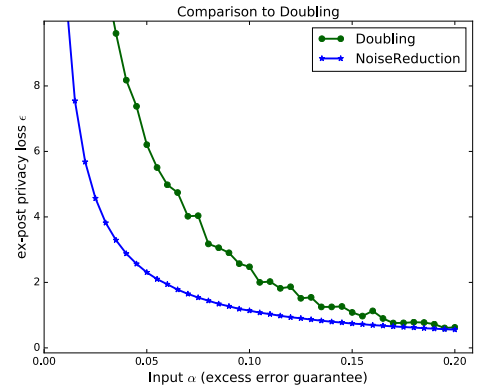
In contrast to our original expectation, the privacy loss due to "testing" hypotheses (the AboveThreshold technique) was significantly larger than that for "generating" them (NoiseReduction). One place the AboveThreshold analysis is loose is in using a theoretical bound on the maximum norm of any hypothesis to compute the sensitivity of queries. The actual norm of hypotheses tested was significantly lower which, if taken as guidance to the practitioner in advance, would drastically improve the privacy guarantee of both adaptive methods.



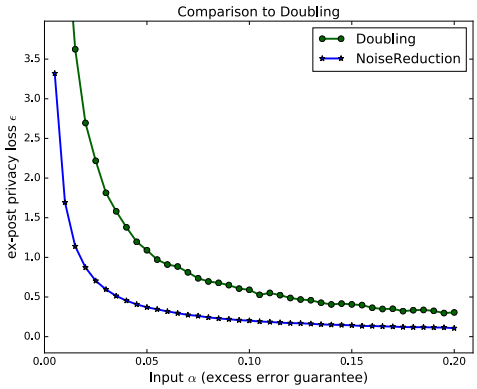
(a) **Linear (ridge) regression, vs theory approach.**



(b) **Regularized logistic regression, vs theory approach.**



(c) **Linear (ridge) regression, vs naive approach.**



(d) **Regularized logistic regression, vs naive approach.**

Figure 1: **Ex-post privacy loss.** (1a) and (1c), left, represent ridge regression on the Twitter dataset, where NoiseReduction and Doubling both use Covariance Perturbation. (1b) and (1d), right, represent logistic regression on the KDD-99 Cup dataset, where both NoiseReduction and Doubling use Output Perturbation. The top plots compare NoiseReduction to the "theory approach": running the algorithm once using the value of ε that guarantees the desired expected error via a utility theorem. The bottom compares to the Doubling baseline. Note the top plots are generous to the theory approach: the theory curves promise only expected error, whereas NoiseReduction promises a high probability guarantee.

CHAPTER 4

How to Use Heuristics for Differential Privacy

4.1. Introduction

Differential privacy is compatible with a tremendous number of powerful data analysis tasks, including essentially any statistical learning problem Kasiviswanathan et al. (2011); Chaudhuri et al. (2011); Bassily et al. (2014) and the generation of synthetic data consistent with exponentially large families of statistics Blum et al. (2013); Roth and Roughgarden (2010); Hardt and Rothblum (2010); Gupta et al. (2012); Nikolov et al. (2013). Unfortunately, it is also beset with a comprehensive set of computational hardness results. Of course, it inherits all of the computational hardness results from the (non-private) agnostic learning literature: for example, even the simplest learning tasks — like finding the best conjunction or linear separator to approximately minimize classification error — are hard Feldman et al. (2009a, 2012a); Diakonikolas et al. (2011a). In addition, tasks that are easy absent privacy constraints can become hard when these constraints are added. For example, although information theoretically, it is possible to privately construct synthetic data consistent with all d-way marginals for d-dimensional data, privately constructing synthetic data for even 2-way marginals is computationally hard Ullman and Vadhan (2010). These hardness results extend even to providing numeric answers to more than quadratically many statistical queries Ullman (2016).

How should we proceed in the face of pervasive computational hardness? We might take inspiration from machine learning, which has not been slowed, despite the fact that its most basic problems (e.g. learning linear separators) are already hard even to approximate. Instead, the field has employed heuristics with tremendous success — including exact optimization of convex surrogate loss functions (as in the case of SVMs), decision tree heuristics, gradient based methods for differentiable but non-convex problems (as in back-propogation for training neural networks), and integer programming solvers (as in recent

work on interpretable machine learning Ustun and Rudin (2016)). Other fields such as operations research similarly have developed sophisticated heuristics including integer program solvers and SAT solvers that are able to routinely solve problems that are hard in the worst case.

The case of private data analysis is different, however. If we are only concerned with performance (as is the case for most machine learning and combinatorial optimization tasks), we have the freedom to try different heuristics, and evaluate our algorithms in practice. Thus the design of heuristics that perform well in practice can be undertaken as an empirical science. In contrast, differential privacy is an inherently worst-case guarantee that cannot be evaluated empirically (see Gilbert and McMillan (2018) for lower bounds for black-box testing of privacy definitions).

In this paper, we build a theory for how to employ *non-private* heuristics (of which there are many, benefitting from many years of intense optimization) to solve computationally hard problems in differential privacy. Our goal is to guide the design of practical algorithms about which we can still prove theorems:

1. We will aim to prove accuracy theorems *under the assumption that our heuristics solve some non-private problem optimally*. We are happy to make this assumption when proving our accuracy theorems, because accuracy is something that can be empirically evaluated on the datasets that we are interested in. An assumption like this is also necessary, because we are designing algorithms for problems that are computationally hard in the worst case. However:

2. We aim to prove that our algorithms are differentially private in the worst case, even under the assumption that our heuristics might fail in an adversarial manner.

4.1.1. Overview of Our Results

Informally, we give a collection of results showing the existence of *oracle-efficient* algorithms for privately solving learning and synthetic data generation problems defined by discrete classes of functions $\mathcal{Q}$ that have a special (but common) combinatorial structure. One might initially ask whether it is possible to give a direct reduction from a non-private but efficient algorithm for solving a learning problem to an efficient private algorithm for solving the same learning problem *without requiring any special structure at all*. However, this is impossible, because there are classes of functions (namely those that have finite VC-dimension but infinite Littlestone dimension) that are known to be learnable absent the constraint of privacy, but are not privately learnable in an information-theoretic sense Bun et al. (2015); Alon et al. (2018). The main question we leave open is whether *being information theoretically learnable under the constraint of differential privacy* is sufficient for oracle-efficient private learning. We give a barrier result suggesting that it might not be.

Before we summarize our results in more detail, we give some informal definitions.

Definitions

We begin by defining the kinds of *oracles* that we will work with, and end-goals that we will aim for. We will assume the existence of oracles for (non-privately) solving learning problems: for example, an oracle which can solve the empirical risk minimization problem for discrete linear threshold functions. Because ultimately oracles will be implemented using heuristics, we consider two types of oracles:

1. *Certifiable* heuristic oracles might fail, but when they succeed, they come with a certificate of success. Many heuristics for solving integer programs are certifiable, including cutting planes methods and branch and bound methods. SAT Solvers (and any other heuristic for solving a decision problem in NP) are also certifiable.

2. On the other had, some heuristics are *non-certifiable*. These heuristics might produce incorrect answers, without any indication that they have failed. Support vector machines and logistic regression are examples of non-certifiable heuristic oracles for learning linear threshold functions.

We define an oracle-efficient *non-robustly* differentially private algorithm to be an algorithm that runs in polynomial time in all relevant parameters given access to an oracle for some problem, and has an accuracy guarantee and a differential privacy guarantee which may both be *contingent* on the guarantees of the oracle — i.e. if the oracle is replaced with a heuristic, the algorithm may no longer be differentially private. Although in certain situations (e.g when we have very high confidence that our heuristics actually do succeed on all instances we will ever encounter) it might be acceptable to have a privacy guarantee that is contingent on having an infallible oracle, we would much prefer a privacy guarantee that held in the worst case. We say that an oracle-efficient algorithm is *robustly* differentially private if its privacy guarantee is not contingent on the behavior of the oracle, and holds in the worst case, even if an adversary is in control of the heuristic that stands in for our oracle.

Learning and Optimization

Our first result is a reduction from efficient non-private learning to efficient private learning over any class of functions $\mathcal{Q}$ that has a small universal identification set Goldman et al. (1993). A universal identification set of size m is a set of m examples such that the labelling of these examples by a function $q \in \mathcal{Q}$ is enough to uniquely identify q. Equivalently, a universal identification set can be viewed as a *separator set* Syrgkanis et al. (2016): for any pair of functions $q \neq q' \in \mathcal{Q}$, there must be some example x in the universal identification set such that $q(x) \neq q(x')$. We will use these terms interchangeably throughout the paper. We show that if $\mathcal{Q}$ has a universal identification set of size m, then given an oracle which solves the empirical risk minimization problem (non-privately) over $\mathcal{Q}$, there is an ε-differentially

private algorithm with additional running time scaling linearly with m and error scaling linearly with m^2/ε that solves the private empirical risk minimization problem over $\mathcal{Q}$. The error can be improved to $O(m^{1.5}\sqrt{\log 1/\delta}/\varepsilon)$, while satisfying (ε, δ)-differential privacy. Many well studied discrete concept classes $\mathcal{Q}$ from the PAC learning literature have small universal identification sets. For example, in d dimensions, boolean conjunctions, disjunctions, parities, and halfspaces defined over the hypercube have universal identification sets of size d. This means that for these classes, our oracle-efficient algorithm has error that is larger than the generic optimal (and computationally inefficient) learner from Kasiviswanathan et al. (2011) by a factor of $O(\sqrt{d})$. Other classes of functions also have small universal identification sets — for example, decision lists have universal identification sets of size d^2.

The reduction described above has the disadvantage that not only its accuracy guarantees — but also its proof of privacy — depend on the oracle correctly solving the empirical risk minimization problem it is given; it is *non-robustly* differentially private. This shortcoming motivates our main technical result: a generic reduction that takes as input any oracle-efficient non-robustly differentially private algorithm (i.e. an algorithm whose privacy proof might depend on the proper functioning of the oracle) and produces an oracle-efficient *robustly* differentially private algorithm, *whenever the oracle is implemented with a certifiable heuristic*. As discussed above, this class of heuristics includes the integer programming algorithms used in most commercial solvers. In combination with our first result, we obtain robustly differentially private oracle-efficient learning algorithms for conjunctions, disjunctions, discrete halfspaces, and any other class of functions with a small universal identification set.

Synthetic Data Generation

We then proceed to the task of constructing synthetic data consistent with a class of queries $\mathcal{Q}$. Following Hsu et al. (2013b); Gaboardi et al. (2014), we view the task of synthetic data

generation as the process of computing an equilibrium of a particular zero sum game played between a data player and a query player. In order to compute this equilibrium, we need to be able to instantiate two objects in an oracle-efficient manner:

1. a private *learning* algorithm for $\mathcal{Q}$ (this corresponds to solving the best response problem for the "query player"), and

2. a *no-regret learning algorithm* for a dual class of functions $\mathcal{Q}_{\text{dual}}$ that results from swapping the role of the data element and the query function (this allows the "data player" to obtain a diminishing regret bound in simulated play of the game).

The no-regret learning algorithm need not be differentially private. From our earlier results, we are able to construct an oracle-efficient robustly differentially private learning algorithm for $\mathcal{Q}$ whenever it has a small universal identification set. On the other hand, Syrgkanis et al. Syrgkanis et al. (2016) show how to obtain an oracle-efficient no regret learning algorithm for a class of functions under the same condition. Hence, we obtain an oracle-efficient robustly differentially private synthetic data generation algorithm for any class of functions $\mathcal{Q}$ for which both $\mathcal{Q}$ and $\mathcal{Q}_{\text{dual}}$ have small universal identification sets. Fortunately, this is the case for many interesting classes of functions, including boolean disjunctions, conjunctions, discrete halfspaces, and parity functions. The result is that we obtain oracle-efficient algorithms for generating private synthetic data for all of these classes. We note that the oracle used by the data player need not be certifiable.

A Barrier Result

Finally, we exhibit a barrier to giving oracle-efficient private learning algorithms for *all* classes of functions $\mathcal{Q}$ known to be privately learnable. We identify a class of private learning algorithms called *perturbed empirical risk minimizers* (pERMs) which output the query that *exactly* minimizes some perturbation of their empirical risk on the dataset. This class of algorithms includes the ones we give in this paper, as well as many other differentially private learning algorithms, including the exponential mechanism and report-

noisy-min. We show that any private pERM can be efficiently used as a no-regret learning algorithm with regret guarantees that depend on the scale of the perturbations it uses. This allows us to reduce to a lower bound on the running time of oracle-efficient online learning algorithms due to Hazan and Koren Hazan and Koren (2016). The result is that there exist finite classes of queries $\mathcal{Q}$ such that any oracle-efficient differentially private pERM algorithm must introduce perturbations that are polynomially large in the size of $|Q|$, whereas any such class is information-theoretically privately learnable with error that scales only with $\log|\mathcal{Q}|$.

The barrier implies that *if* oracle-efficient differentially private learning algorithms are as powerful as inefficient differentially private learning algorithms, then these general oracle efficient private algorithms must not be perturbed empirical risk minimizers. We conjecture that the set of problems solvable by oracle-efficient differentially private learners is strictly smaller than the set of problems solvable information theoretically under the constraint of differential privacy, but leave this as our main open question.

4.1.2. Additional Related Work

Conceptually, the most closely related piece of work is the "DualQuery" algorithm of Gaboardi et al. (2014), which in the terminology of our paper is a robustly private oracle-efficient algorithm for generating synthetic data for k-way marginals for constant k. The main idea in Gaboardi et al. (2014) is to formulate the private optimization problem that needs to be solved so that the only computationally hard task is one that does not depend on private data. There are other algorithms that can straightforwardly be put into this framework, like the projection algorithm from Nikolov et al. (2013). This approach immediately makes the privacy guarantees independent of the correctness of the oracle, but significantly limits the algorithm design space. In particular, the DualQuery algorithm (and the oracle-efficient version of the projection algorithm from Nikolov et al. (2013)) has running time that is proportional to $|\mathcal{Q}|$, and so can only handle polynomially sized classes of queries (which is why k needs to be held constant). The main contribution of our paper

is to be able to handle private optimization problems in which the hard computational step is *not* independent of the private data. This is significantly more challenging, and is what allows us to give oracle-efficient robustly private algorithms for constructing synthetic data for exponentially large families $\mathcal{Q}$. It is also what lets give oracle-efficient private *learning* algorithms over exponentially large $\mathcal{Q}$ for the first time.

A recent line of work starting with the "PATE" algorithm Papernot et al. (2016) together with more recent theoretical analyses of similar algorithms by Dwork and Feldman, and Bassily, Thakkar, and Thakurta Dwork and Feldman (2018); Bassily et al. (2018) can be viewed as giving oracle-efficient algorithms for an easier learning task, in which the goal is to produce a finite number of private *predictions* rather than privately output the model that makes the predictions. These can be turned into oracle efficient algorithms for outputting a private model *under the assumption* that the mechanism has access to an additional source of unlabeled data drawn from the same distribution as the private data, but that does not need privacy protections. In this setting, there is no need to take advantage of any special structure of the hypothesis class $\mathcal{Q}$, because the information theoretic lower bounds on private learning proven in Bun et al. (2015); Alon et al. (2018) do not apply. In contrast, our results apply without the need for an auxiliary source of non-private data.

Privately producing *contingency tables*, and synthetic data that encode them — i.e. the answers to statistical queries defined by conjunctions of features — has been a key challenge problem in differential privacy at least since Barak et al. (2007). Since then, a number of algorithms and hardness results have been given Ullman and Vadhan (2010); Gupta et al. (2013); Kasiviswanathan et al. (2010); Thaler et al. (2012); Hardt et al. (2012); Feldman and Kothari (2014); Chandrasekaran et al. (2014). This paper gives the first oracle-efficient algorithm for generating synthetic data consistent with a full contingency table, and the first oracle-efficient algorithm for answering arbitrary conjunctions to near optimal error.

Technically, our work is inspired by Syrgkanis et al. Syrgkanis et al. (2016) who show how a small separator set (equivalently a small universal identification set) can be used to derive

oracle-efficient no-regret algorithms in the contextual bandit setting. The small separator property has found other uses in online learning, including in the oracle-efficient construction of nearly revenue optimal auctions Dudík et al. (2017). Hazan and Koren Hazan and Koren (2016) show lower bounds for oracle-efficient no-regret learning algorithms in the experts setting, which forms the basis of our barrier result. More generally, there is a rich literature studying oracle-efficient algorithms in machine learning Beygelzimer et al. (2005); Balcan et al. (2008); Beygelzimer et al. (2016) and optimization Ben-Tal et al. (2015) as a means of dealing with worst-case hardness, and more recently, for machine learning subject to fairness constraints Agarwal et al. (2018); Kearns et al. (2018); Alabi et al. (2018).

We also make crucial use of a property of differentially private algorithms, first shown by Cummings et al. (2016): That when differentially private algorithms are run on databases of size n with privacy parameter $\varepsilon \approx 1/\sqrt{n}$, then they have similar output distributions when run on datasets that are *sampled from the same distribution*, rather than just on neighboring datasets. In Cummings et al. (2016), this was used as a tool to show the existence of *robustly generalizing* algorithms (also known as *distributionally private* algorithms in Blum et al. (2013)). We prove a new variant of this fact that holds when the datasets are not sampled i.i.d. and use it for the first time in an analysis to prove differential privacy. The technique might be of independent interest.

4.2. Preliminaries

4.2.1. Statistical Queries and Separator Sets

We study learning (optimization) and synthetic data generation problems for statistical queries defined over a data universe $\mathcal{X}$. A statistical query over $\mathcal{X}$ is a function $q : \mathcal{X} \to \{0, 1\}$. A statistical query can represent, e.g. any binary classification model or the binary loss function that it induces. Given a dataset $S \in \mathcal{X}^n$, the value of a statistical query q on S is defined to be $q(S) = \frac{1}{n}\sum_{i=1}^{n} q(S_i)$. In this chapter, we will generally think about query

classes $\mathcal{Q}$ that represent standard *hypothesis classes* from learning theory – like conjunctions, disjunctions, halfspaces, etc.

In this chapter, we will make crucial use of *universal identification sets* for classes of statistical queries. Universal identification sets are equivalent to *separator sets*, defined (in a slightly more general form) in Syrgkanis et al. (2016).

Definition 4.2.1 (Goldman et al. (1993); Syrgkanis et al. (2016))**.** *A set $U \subseteq \mathcal{X}$ is a* universal identification set *or* separator set *for a class of statistical queries $\mathcal{Q}$ if for every pair of distinct queries $q, q' \in \mathcal{Q}$, there is an $x \in U$ such that:*

$$q(x) \neq q(x')$$

If $|U| = m$, then we say that $\mathcal{Q}$ has a separator set of size m.

Many classes of statistical queries defined over the boolean hypercube have separator sets of size proportional to their VC-dimension. For example, boolean conjunctions, disjunctions, halfspaces defined over the hypercube, and parity functions in d dimensions all have separator sets of size d. When we solve learning problems over these classes, we will be interested in the set of queries that define the 0/1 loss function over these classes: but as we observe in Appendix A.2.1, if a hypothesis class has a separator set of size m, then so does the class of queries representing the empirical loss for functions in that hypothesis class.

4.2.2. Learning and Synthetic Data Generation

We study private learning as empirical risk minimization (the connection between in-sample risk and out-of-sample risk is standard, and follows from e.g. VC-dimension bounds Kearns and Vazirani (1994a) or directly from differential privacy (see e.g. Bassily et al. (2014); Dwork et al. (2015b))). Such problems can be cast as finding a function q in a class $\mathcal{Q}$ that minimizes $q(S)$, subject to differential privacy (observe that the empirical risk of a hypothesis is a statistical query — see Appendix A.2.1). We will therefore study minimization problems over classes of statistical queries generally:

Definition 4.2.2. *We say that a randomized algorithm $M : \mathcal{X}^n \to \mathcal{Q}$ is an (α, β)-minimizer for $\mathcal{Q}$ if for every dataset $S \in \mathcal{X}^n$, with probability $1 - \beta$, it outputs $M(S) = q$ such that:*

$$q(S) \leq \arg\min_{q^* \in \mathcal{Q}} q^*(S) + \alpha$$

Synthetic data generation, on the other hand, is the problem of constructing a *new* dataset $\hat{S}$ that approximately agrees with the original dataset with respect to a fixed set of statistical queries:

Definition 4.2.3. *We say that a randomized algorithm $M : \mathcal{X}^n \to \mathcal{X}^*$ is an (α, β)-accurate synthetic data generation algorithm for $\mathcal{Q}$ if for every dataset $S \in \mathcal{X}^n$, with probability $1 - \beta$, it outputs $M(S) = \hat{S}$ such that for all $q \in \mathcal{Q}$:*

$$|q(S) - q(\hat{S})| \leq \alpha$$

4.2.3. Robust Differential Privacy

For the definitions of oracle-efficiency, certifiable oracle-efficiency, and oracle-dependency used through the paper, see subsection 2.2. With these definitions in hand, we are able to define the notion of robust differential privacy in the context of oracle-dependent algorithms. If our algorithms are merely *oracle equivalent* to differentially private algorithms, then their privacy guarantees depend on the correctness of the oracle. However, we would prefer that the *privacy* guarantee of the algorithm not depend on the success of the oracle. We call such algorithms *robustly* differentially private.

Definition 4.2.4. *An oracle-efficient algorithm M is (ε, δ)-robustly differentially private if it satisfies (ε, δ)-differential privacy even under worst-case performance of a heuristic optimization oracle. In other words, it is differentially private for every heuristic oracle $\mathcal{O}$ that it might be instantiated with.*

We write that an oracle efficient algorithm is non-robustly differentially private to mean that it is oracle equivalent to a differentially private algorithm. Naturally any robustly differentially private algorithm is non-robustly differentially private.

4.3. Oracle Efficient Optimization

In this section, we show how weighted optimization oracles can be used to give differentially private oracle-efficient optimization algorithms for many classes of queries with performance that is worse only by a $\sqrt{d}$ factor compared to that of the (computationally inefficient) exponential mechanism. The first algorithm we give is not robustly differentially private — that is, its differential privacy guarantee relies on having access to a perfect oracle. We then show how to make that algorithm (or any other algorithm that is oracle equivalent to a differentially private algorithm) robustly differentially private when instantiated with a certifiable heuristic optimization oracle.

4.3.1. A (Non-Robustly) Private Oracle Efficient Algorithm

In this section, we give an oracle-efficient (non-robustly) differentially private optimization algorithm that works for any class of statistical queries that has a small separator set. Intuitively, it is attempting to implement the "Report-Noisy-Min" algorithm (see e.g. Dwork and Roth (2014b)), which outputs the query q that minimizes a (perturbed) estimate $\hat{q}(S) \equiv q(S) + Z_q$ where $Z_q \sim \text{Lap}(1/\varepsilon)$ for each $q \in \mathcal{Q}$. Because Report-Noisy-Min samples an independent perturbation for each query $q \in \mathcal{Q}$, it is inefficient: its run time is linear in $|\mathcal{Q}|$. Our algorithm – "Report Separator-Perturbed Min" (RSPM) – instead augments the dataset S in a way that implicitly induces perturbations of the query values $q(S)$. The perturbations are no longer independent across queries, and so to prove privacy, we need to use the structure of a separator set.

The algorithm is straightforward: it simply augments the dataset with one copy of each element of the separator set, each with a weight drawn independently from the Laplace distribution. All original elements in the dataset are assigned weight 1. The algorithm then

simply passes this weighted dataset to the weighted optimization oracle, and outputs the resulting query. The number of random variables that need to be sampled is therefore now equal to the size of the separator set, instead of the size of $\mathcal{Q}$. The algorithm is closely related to a no-regret learning algorithm given in Syrgkanis et al. (2016) — the only difference is in the magnitude of the noise added, and in the analysis, since we need a substantially stronger form of stability.

Report Separator-Perturbed Min (RSPM)
Given: A separator set $U = \{e_1, \ldots, e_m\}$ for a class of statistical queries $\mathcal{Q}$, a weighted optimization oracle $\mathcal{O}^*$ for $\mathcal{Q}$, and a privacy parameter ε.
Input: A dataset $S \in \mathcal{X}^n$ of size n.
Output: A statistical query $q \in \mathcal{Q}$.

Sample $\eta_i \sim Lap(2m/\varepsilon)$ for $i \in \{1, \ldots, m\}$
Construct a weighted dataset WD of size $n + m$ as follows:

$$WD(S, \eta) = \{(x_i, 1) : x_i \in S\} \cup \{(e_i, \eta_i) : e_i \in U\}$$

Output $q = \mathcal{O}^*(WD(S, \eta))$.

It is thus immediate that the Report Separator-Perturbed Min algorithm is oracle-efficient whenever the size of the separator set m is polynomial: it simply augments the dataset with a single copy of each of m separator elements, makes m draws from the Laplace distribution, and then makes a single call to the oracle:

Theorem 4.3.1. *The Report Separator-Perturbed Min algorithm is oracle-efficient.*

The accuracy analysis for the Report Separator-Perturbed Min algorithm is also straightforward, and follows by bounding the weighted sum of the additional entries added to the original data set.

Theorem 4.3.2. *The Report Separator-Perturbed Min algorithm is an (α, β)-minimizer for $\mathcal{Q}$ for:*

$$\alpha = \frac{4m^2 \log(m/\beta)}{\varepsilon n}$$

Proof. Let q' be the query returned by RSPM, and let q^* be the true minimizer $q^* = \arg\min_{q \in \mathcal{Q}} q^*(S)$. Then we show that with probability $1 - \beta, q'(S) \leq q^*(S) + \alpha$. By the CDF of

the Laplace distribution and a union bound over the m random variables η_i, we have that with probability $1-\beta$:

$$\forall i,\ |\eta_i| \leq \frac{2m\log(m/\beta)}{\varepsilon}.$$

Since for every query q, $q(e_i) \in [0,1]$, this means that with probability $1-\beta$, $q'(WD) \geq q'(S) - m \cdot \frac{2m\log(m/\beta)}{\varepsilon n}$. Similarly $q^*(WD) \leq q^*(S) + 2m \cdot \frac{m\log(m/\beta)}{\varepsilon n}$. Combining these bounds gives:

$$q'(S) \leq q'(WD) + 2m^2\frac{\log(m/\beta)}{\varepsilon n} \leq q^*(WD) + 2m^2\frac{\log(m/\beta)}{\varepsilon n} \leq q^*(S) + \frac{4m^2\log(m/\beta)}{\varepsilon n}$$

as desired, where the second inequality follows because by definition, q' is the true minimizer on the weighted dataset WD. □

Remark 4.3.3. *We can bound the expected error of RSPM using Theorem 4.3.2 as well. If we denote the error of RSPM by E, we've shown that for all β, $\Pr\left[E \geq \frac{4m^2\log(m/\beta)}{\varepsilon n}\right] \leq \beta$. Thus $\Pr\left[\frac{\varepsilon n E}{4m^2} - \log m \geq \log(1/\beta)\right] \leq \beta$ for all β. Let $\tilde{E} = \max(0, \frac{\varepsilon n E}{4m^2} - \log m)$. Since $\tilde{E}$ is non-negative:*

$$\mathbb{E}\left[\tilde{E}\right] = \int_0^\infty \Pr\left[\tilde{E} \geq t\right] \leq \int_0^\infty e^{-t} = 1.$$

Hence $\frac{\varepsilon n\,\mathbb{E}[E]}{4m^2} - \log m \leq \mathbb{E}\left[\tilde{E}\right] \leq 1$, and so $\mathbb{E}[E] \leq \frac{4m^2}{\varepsilon n}(1+\log m)$.

The privacy analysis is more delicate, and relies on the correctness of the oracle.

Theorem 4.3.4. *If $\mathcal{O}^*$ is a weighted optimization oracle for $\mathcal{Q}$, then the Report Separator-Perturbed Min algorithm is ε-differentially private.*

Proof. We begin by introducing some notation. Given a weighted dataset $WD(S,\eta)$, and a query $q \in \mathcal{Q}$, let $q(S,\eta) = q(S) + \sum_{e_i \in U} q(e_i)\eta_i$ be the value when q is evaluated on the weighted dataset given the realization of the noise η. To allow us to distinguish queries that are output by the algorithm on different datasets and different realizations of the

perturbations, write $\mathcal{Q}(S,\eta) = \mathcal{O}^*(WD(S,\eta))$. Fix any $q \in \mathcal{Q}$, and define:

$$\mathcal{E}(q,S) = \{\eta : \mathcal{Q}(S,\eta) = q\}$$

to be the event defined on the perturbations η that the mechanism outputs query q. Given a fixed $q \in \mathcal{Q}$ we define a mapping $f_q(\eta) : \mathbb{R}^m \to \mathbb{R}^m$ on noise vectors as follows:

1. If $q(e_i) = 1, f_q(\eta)_i = \eta_i - 2$
2. If $q(e_i) = 0, f_q(\eta)_i = \eta_i + 2$

Equivalently, $f_q(\eta)_i = \eta_i + 2(1 - 2q(e_i))$.

We now make a couple of observations about the function f_q.

Lemma 4.3.5. *Fix any $\hat{q} \in \mathcal{Q}$ and any pair of neighboring datasets S, S'. Let $\eta \in \mathcal{E}(\hat{q}, S)$ be such that $\hat{q}$ is the unique minimizer $\hat{q} \in \inf_{q \in \mathcal{Q}} q(S,\eta)$. Then $f_{\hat{q}}(\eta) \in \mathcal{E}(\hat{q}, S')$. In particular, this implies that for any such η:*

$$\mathbb{1}(\eta \in \mathcal{E}(\hat{q}, S)) \leq \mathbb{1}(f_{\hat{q}}(\eta) \in \mathcal{E}(\hat{q}, S'))$$

Proof. For this argument, it will be convenient to work with *un-normalized* versions of our queries, so that $q(S) = \sum_{x_i \in S} q(x_i)$ — i.e. we do not divide by the dataset size n. Note that this change of normalization does not change the identity of the minimizer. Under this normalization, the queries q are now 1-sensitive, rather than $1/n$ sensitive.

Recall that $\mathcal{Q}(S,\eta) = \hat{q}$. Suppose for point of contradiction that $\mathcal{Q}(S', f_{\hat{q}}(\eta)) = \tilde{q} \neq \hat{q}$. This in particular implies that $\tilde{q}(S', f_{\hat{q}}(\eta)) \leq \hat{q}(S', f_{\hat{q}}(\eta))$.

We first observe that $\hat{q}(S',\eta) - \tilde{q}(S',\eta) < 2$. This follows because:

$$\tilde{q}(S',\eta) \geq \tilde{q}(S,\eta) - 1 > \hat{q}(S,\eta) - 1 \geq \hat{q}(S',\eta) - 2 \tag{4.1}$$

Here the first inequality follows because the un-normalized queries q are 1-sensitive, the second follows because $\hat{q} \in \arg\min_{q \in \mathcal{Q}} q(S, \eta)$ is the unique minimizer, and the last inequality follows from the fact that S and S' are neighbors.

Next, we write:

$$\tilde{q}(S', f_{\hat{q}}(\eta)) - \hat{q}(S', f_{\hat{q}}(\eta)) = \tilde{q}(S', \eta) - \hat{q}(S', \eta) + \sum_{i=1}^{m} (\tilde{q}(e_i) - \hat{q}(e_i))(f_{\hat{q}}(\eta_i) - \eta_i)$$

Consider each term in the final sum: $(\tilde{q}(e_i) - \hat{q}(e_i))(f_{\hat{q}}(\eta_i) - \eta_i)$. Observe that by construction, each of these terms is non-negative: Clearly if $\tilde{q}(e_i) = \hat{q}(e_i)$, then the term is 0. Further, if $\tilde{q}(e_i) \neq \hat{q}(e_i)$, then by construction, $(\tilde{q}(e_i) - \hat{q}(e_i))(f_{\hat{q}}(\eta_i) - \eta_i) = 2$. Finally, by the definition of a separator set, we know that there is at least one index i such that $\tilde{q}(e_i) \neq \hat{q}(e_i)$. Thus, we can conclude:

$$\tilde{q}(S', f_{\hat{q}}(\eta)) - \hat{q}(S', f_{\hat{q}}(\eta)) \geq \tilde{q}(S', \eta) - \hat{q}(S', \eta) + 2 > 0$$

where the final inequality follows from applying inequality (4.1). But rearranging, this means that $\hat{q}(S', f_{\hat{q}}(\eta)) < \tilde{q}(S', f_{\hat{q}}(\eta))$, which contradicts the assumption that $\mathcal{Q}(S', f_{\hat{q}}(\eta)) = \tilde{q}$. □

Let p denote the probability density function of the joint distribution of the Laplace random variables η, and by abuse of notation also of each individual η_i.

Lemma 4.3.6. *For any $r \in \mathbb{R}^m, q \in \mathcal{Q}$:*

$$p(\eta = r) \leq e^{\varepsilon} p(\eta = f_q(r))$$

Proof. For any index i and $z \in \mathbb{R}$, we have $p(\eta_i = z) = \frac{\varepsilon}{4m} e^{-|z|\varepsilon/(2m)}$. In particular, if $|x - y| \leq 2$, $p(\eta_i = y) \leq e^{\varepsilon/m} p(\eta_i = x)$. Since for all i and $r \in \mathbb{R}^m$ $|f_q(r)_i - r_i| \leq 1$, we have:

$$\frac{p(\eta = f_q(r))}{p(\eta = r)} = \prod_{i=1}^{m} \frac{p(\eta_i = f_q(r)_i)}{p(\eta_i = r_i)} \leq \prod_{i=1}^{m} e^{\varepsilon/m} = e^{\varepsilon}.$$

□

Lemma 4.3.7. *Fix any class of queries $\mathcal{Q}$ that has a finite separator set $U = \{e_1, \ldots, e_m\}$. For every dataset S there is a subset $B \subseteq \mathbb{R}^m$ such that:*

1. $\Pr[\eta \in B] = 0$ *and*

2. *On the restricted domain $\mathbb{R}^m \setminus B$, there is a unique minimizer $q' \in \arg\min_{q \in \mathcal{Q}} q(S, \eta)$*

Proof. Let:

$$B = \{\eta : \left| \arg\min_{q \in \mathcal{Q}} (q(S) + \sum_{i=1}^{m} \eta_i q(e_i)) \right| > 1\}$$

be the set of η values that do *not* result in unique minimizers q'.

Because $\mathcal{Q}$ is a finite set[1], by a union bound it suffices to show that for any two distinct queries $q_1, q_2 \in \mathcal{Q}$,

$$\Pr_{\eta}\left[q_1(S) + \sum_{i=1}^{m} \eta_i q_1(e_i) = q_2(S) + \sum_{i=1}^{m} \eta_i q_2(e_i) \right] = 0.$$

This follows from the continuity of the Laplace distribution. Let i be any index such that $q_1(e_i) \neq q_2(e_i)$ (recall that by the definition of a separator set, such an index is guaranteed to exist). For any fixed realization of $\{\eta_j\}_{j \neq i}$, there is a single value of η_i that equalizes $q_1(S, \eta)$ and $q_2(S, \eta)$. But any single value is realized with probability 0.

□

[1] Any class of queries $\mathcal{Q}$ with a separator set of size m can be no larger than 2^m.

We now have enough to complete the proof. We have for any query $\hat{q}$:

$$\begin{aligned}
\Pr[RSPM(S) = \hat{q}] &= \Pr[\eta \in \mathcal{E}(\hat{q}, S)] \\
&= \int_{\mathbb{R}^m} p(\eta)\mathbb{1}(\eta \in \mathcal{E}(\hat{q}, S))d\eta \\
&= \int_{\mathbb{R}^m \setminus B} p(\eta)\mathbb{1}(\eta \in \mathcal{E}(\hat{q}, S))d\eta && B \text{ has 0 measure. (Lemma 4.3.7)} \\
&\leq \int_{\mathbb{R}^m \setminus B} p(\eta)\mathbb{1}(f_{\hat{q}}(\eta) \in \mathcal{E}(\hat{q}, S'))d\eta && \text{Lemma 4.3.7} \implies \text{Lemma 4.3.5} \\
&\leq \int_{\mathbb{R}^m \setminus B} e^{\varepsilon} p(f_{\hat{q}}(\eta))\mathbb{1}(f_{\hat{q}}(\eta) \in \mathcal{E}(\hat{q}, S'))d\eta && \text{Lemma 4.3.6} \\
&\leq \int_{\mathbb{R}^m \setminus f_{\hat{q}}(B)} e^{\varepsilon} p(\eta)\mathbb{1}(\eta \in \mathcal{E}(\hat{q}, S'))\left|\frac{\partial f_{\hat{q}}}{\partial \eta}\right| d\eta && \text{Change of variables } \eta \to f_{\hat{q}}(\eta) \\
&= \int_{\mathbb{R}^m} e^{\varepsilon} p(\eta)\mathbb{1}(\eta \in \mathcal{E}(\hat{q}, S'))d\eta && f_{\hat{q}}(B) \text{ has 0 measure, } \left|\frac{\partial f_{\hat{q}}}{\partial \eta}\right| = 1 \\
&= e^{\varepsilon} \Pr[\eta \in \mathcal{E}(\hat{q}, S')] \\
&= e^{\varepsilon} \Pr[RSPM(S') = \hat{q}]
\end{aligned}$$

□

In Appendix A.2.2, we give a somewhat more complicated analysis to show that by using Gaussian perturbations rather than Laplace perturbations, it is possible to improve the accuracy of the RSPM algorithm by a factor of $\sqrt{m}$, at the cost of satisfying (ε, δ)-differential privacy:

Theorem 4.3.8. *The Gaussian RSPM algorithm is (ε, δ)-differentially private, and is an oracle-efficient (α, β)-minimizer for any class of functions $\mathcal{Q}$ that has a universal identifications sequence of size m for:*

$$\alpha = O\left(\frac{m\sqrt{m\ln(m/\beta)\ln(1/\delta)}}{\varepsilon n}\right)$$

See Appendix A.2.2 for the algorithm and its analysis.

It is instructive to compare the accuracy that we can obtain with oracle-efficient algorithms to the accuracy that can be obtained via the (inefficient, and generally optimal) exponential mechanism based generic learner from Kasiviswanathan et al. (2011). The existence of a universal identification set for $\mathcal{Q}$ of size m implies $|\mathcal{Q}| \leq 2^m$ (and for many interesting classes of queries, including conjunctions, disjunctions, parities, and discrete halfspaces over the hypercube, this is an equality — see Appendix A.2.1). Thus, the exponential-mechanism based learner from Kasiviswanathan et al. (2011) is (α, β)-accurate for:

$$\alpha = O\left(\frac{m + \log(1/\beta)}{\varepsilon n}\right).$$

Comparing this bound to ours, we see that we can obtain oracle-efficiency at a cost of roughly a factor of $\sqrt{m}$ in our error bound. Whether or not this cost is necessary is an interesting open question.

We can conclude that for a wide range of hypothesis classes $\mathcal{Q}$ including boolean conjunctions, disjunctions, decision lists, discrete halfspaces, and several families of circuits of logarithmic depth (see Appendix A.2.1) there is an oracle-efficient differentially private learning algorithm that obtains accuracy guarantees within small polynomial factors of the optimal guarantees of the (inefficient) exponential mechanism.

4.3.2. A Robustly Differentially Private Oracle-Efficient Algorithm

The RSPM algorithm is not *robustly* differentially private, because its privacy proof depends on the oracle succeeding. This is an undesirable property for RSPM and other algorithms like it, because we do not expect to have access to *actual* oracles for hard problems even if we expect that there are certain families of problems for which we can reliably solve typical instances[2]. In this section, we show how to remedy this: we give a black box reduction, starting from a (non-robustly) differentially private algorithm $\mathcal{A}_{\mathcal{O}}$ that is implemented using

[2]There may be situations in which it is acceptable to use non robustly differentially private oracle-efficient algorithms — for example, if the optimization oracle is so reliable that it has never been observed to fail on the domain of interest. But robust differential privacy provides a worst-case guarantee which is preferable.

a *certifiable* heuristic[3] oracle $\mathcal{O}$, and producing a robustly differentially private algorithm $\tilde{\mathcal{A}}_{\mathcal{O}}$ for solving the same problem. $\tilde{\mathcal{A}}_{\mathcal{O}}$ will be (ε, δ)-differentially private for a parameter δ that we may choose, and will have a factor of roughly $\tilde{O}(1/\delta)$ running time overhead on top of $\mathcal{A}_{\mathcal{O}}$. So if $\mathcal{A}_{\mathcal{O}}$ is oracle efficient, so is $\tilde{\mathcal{A}}_{\mathcal{O}}$ whenever the chosen value of $\delta \geq 1/\text{poly}(n)$. If the oracle never fails, then we can prove utility guarantees for it when $\mathcal{A}_{\mathcal{O}}$ has such guarantees, since it just runs $\mathcal{A}_{\mathcal{O}}$ (using a smaller privacy parameter) on a random sub-sample of the original dataset. But the privacy guarantees hold even in the worst case of the behavior of the oracle. We call this reduction the *Private Robust Subsampling Meta Algorithm* or **PRSMA**.

Private Robust Subsampling Meta Algorithm (PRSMA)
Given: Privacy parameters $\varepsilon, \delta \geq 0$ and an oracle-efficient differentially private algorithm $\mathcal{A}^{\varepsilon}_{\mathcal{O}} : \mathcal{X}^n \to \mathcal{M}$, implemented with a certifiable heuristic oracle $\mathcal{O}$.
Input: A dataset $S \in \mathcal{X}^n$ of size n.
Output: An output $m \in \mathcal{M}$ or $\perp$ ("Fail").

1: Randomly partition S into $K = \frac{1}{\varepsilon}(1 + \log(\frac{2}{\delta}))$ equally sized datasets $\{S_i\}_{i=1}^{K}$. (If n is not divisible by K, first discard $n \mod K$ elements at random.)
2: **for** $i = 1 \ldots K$ **do**
3: Set $o_i = PASS$
4: **for** $t = 1 \ldots \frac{\log(K/\delta)}{\delta}$ **do**
5: Compute $\mathcal{A}^{\varepsilon'}_{\mathcal{O}}(S_i) = a_{it}$, where $\varepsilon' = \frac{1}{\sqrt{8\frac{n}{K}\log(2K/\delta)}}$
6: If $a_{it} = \perp$, set $o_i = \perp$
7: Compute $T = \#\{o_i \neq \perp\}$. Let $\tilde{T} = T + z$, where $z \sim \text{Lap}(\frac{1}{\varepsilon})$.
8: Test if $\tilde{T} > \frac{1}{\varepsilon}(1 + \log(\frac{1}{\delta}))$, if no output $\perp$ and halt. **Else:**
9: Sample a uniformly at random from $\{a_{it} : o_i \neq \perp\}$.
10: Output a.

Intuition and Proof Outline

Before we describe the analysis of **PRSMA**, a couple of remarks are helpful in order to set the stage.

1. At first blush, one might be tempted to assert that if an oracle-efficient non-robustly differentially private algorithm is implemented using a certifiable heuristic oracle, then it will sample from a differentially private distribution *conditioned on the event*

[3] We recall that heuristics for solving integer programs (such as cutting planes methods, branch and bound, and branch and cut methods, as implemented in commercial solvers) and SAT solvers are certifiable.

that the heuristic oracle doesn't fail. But a moment's thought reveals that this isn't so: the possibility of failures both on the original dataset S and on the (exponentially many) neighboring datasets S' can substantially change the probabilities of arbitrary events Ω, and how these probabilities differ between neighboring datasets.

2. Next, one might think of the following simple candidate solution: Run the algorithm $\mathcal{A}_{\mathcal{O}}(S)$ roughly $\tilde{O}(1/\delta)$ many times in order to check that the failure probability of the heuristic algorithm on S is $\ll \delta$, and then output a sample of $\mathcal{A}_{\mathcal{O}}(S)$ only if this is so. But this doesn't work either: the failure probability itself will change if we replace S with a neighboring dataset S', and so this won't be differentially private. In fact, there is no reason to think that the failure probability of $\mathcal{A}_{\mathcal{O}}$ will be a low sensitivity function of S, so there is no way to privately estimate the failure probability to non-trivial error.

It *is* possible to use the *subsample-and-aggregate* procedure of Nissim et al. (2007) to randomly partition the dataset into K pieces S_i, and privately estimate on *how many* of these pieces $\mathcal{A}_{\mathcal{O}}(S_i)$ fails with probability $\ll \delta$. The algorithm can then then fail if this private count is not sufficiently large. In fact, this is the first thing that **PRSMA** does, in lines 1-10, setting $o_i = PASS$ for those pieces S_i such that it seems that the probability of failure is $\ll \delta$, and setting $o_i = \perp$ for the others.

But the next step of the algorithm is to randomly select one of the partition elements S_i amongst the set that passed the earlier test: i.e. amongst the set such that $o_i \neq \perp$ — and return one of the outputs a that had been produced by running $\mathcal{A}_{\mathcal{O}}(S_i)$. It is not immediately clear why this should be private, because *which* partition elements passed the test $\{i : o_i \neq \perp\}$ is not itself differentially private. Showing that this results in a differentially private output is the difficult part of the analysis.

To get an idea of the problem that we need to overcome, consider the following situation which our analysis must rule out: Fix a partition of the dataset $S_1, \ldots, S_K$, and imagine that

each partition element passes: we have $o_i \neq \perp$ for all i. Now suppose that there is some event Ω such that $\Pr[\mathcal{A}_{\mathcal{O}}(S_1) \in \Omega] \geq 1/2$, but $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega]$ is close to 0 for all $i \neq 1$. Since $K \approx 1/\varepsilon$, and the final output is drawn from a uniformly random partition element, this means that **PRSMA** outputs an element of Ω with probability $\Omega(\varepsilon)$. Suppose that on a neighboring dataset S', S_1 no longer passes the test and has $o_1 = \perp$. Since it is no longer a candidate to be selected at the last step, we now have that on S', **PRSMA** outputs an element of Ω with probability close to 0. This is a violation of (ε, δ)-differential privacy for any non-trivial value of δ (i.e. $\delta \leq O(\varepsilon)$).

The problem is that (fixing a partition of S into $S_1, \ldots, S_K$) moving to a neighboring dataset S' can potentially arbitrarily change the probability that any single element S_i survives to step 11 of the algorithm, which can in principle change the probability of arbitrary events Ω by an *additive* $\pm O(\varepsilon)$ term, rather than a *multiplicative* $1 \pm O(\varepsilon)$ *factor*.

Since we are guaranteed that (with high probability) if we make it to step 11 without failing, then at least $\Omega(1/\varepsilon)$ elements S_i have survived with $o_i \neq \perp$, it would be sufficient for differential privacy if for every event Ω, the probabilities $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega]$ were within a constant factor of each other, for all i. Then a change of whether a single partition element S_i survives with $o_i \neq \perp$ or not would only add or remove an ε *fraction* of the total probability mass on event Ω. While this seems like a "differential-privacy" like property, but it is not clear that the fact that $\mathcal{A}_{\mathcal{O}^*}$ is differentially private can help us here, because the partition elements S_i, S_j are not neighboring datasets — in fact, they are disjoint. But as we show, it does in fact guarantee this property *if* we set the privacy parameter ε' to be sufficiently small — to roughly $O(1/\sqrt{n/K})$ in step 5.

With this intuition setting the stage, the roadmap of the proof is as follows. For notational simplicity, we write $\mathcal{A}(\cdot)$ to denote $\mathcal{A}_{\mathcal{O}^*}(\cdot)$, the oracle-efficient algorithm when implemented with a perfect oracle.

1. We observe that ε-differential privacy implies that the log-probability of any event Ω when $\mathcal{A}(\cdot)$ is run on S_i changes by less than an additive factor of ε when an element of S_i is changed. We use a method of bounded differences argument to show that this implies that the log-probability density function concentrates around its expectation, where the randomness is over the subsampling of S_i from S. A similar result is proven in Cummings et al. (2016) to show that differentially private algorithms achieve what they call "perfect generalization." We need to prove a generalization of their result because in our case, the elements of S_i are not selected independently of one another. This guides our choice of ε' in step 5 of the algorithm. (Lemma 4.3.10)

2. We show that with high probability, for every S_i such that $o_i \neq \perp$ after step 10 of the algorithm, $\mathcal{A}_{\mathcal{O}}(S_i)$ fails with probability at most $O(\delta)$. By Lemma 2.2.6, this implies that it is δ-close in total variation distance to $\mathcal{A}(S_i)$.

3. We observe that fixing a partition, on a neighboring dataset, only one of the partition elements S_i changes — and hence changes its probability of having $o_i \neq \perp$. Since with high probability, conditioned on **PRSMA** not failing, $\Omega(1/\varepsilon)$ partition elements survive with $o_i \neq \perp$, parts 1 and 2 imply that changing a single partition element S_i only changes the probability of realizing any outcome event by a *multiplicative* factor of $\approx 1 + \varepsilon$.

The Main Theorem

Theorem 4.3.9. ***PRSMA*** *is* (ε, δ) *differentially private when given as input:*

1. *An oracle-efficient non-robustly differentially private algorithm* $\mathcal{A}_{\mathcal{O}}$ *implemented with a certifiable heuristic oracle* $\mathcal{O}$*, and*

2. *Privacy parameters* $(\varepsilon^*, \delta^*)$ *where* $\varepsilon^* = \frac{\varepsilon}{62} \leq \frac{1}{2}$ *and* $\delta^* = \frac{\delta}{11} \leq \frac{1}{2}$.

Proof. We analyze **PRSMA** with privacy parameters ε and δ, optimizing the constants at the end. Fix an input dataset S with $|S| = n$, and an adjacent dataset $S' \sim S$, such that without loss of generality S, S' differ in the element $x_1 \neq x_1'$. We denote the **PRSMA** routine with input $\mathcal{A}_{\mathcal{O}}$ and dataset S by $\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S})$. We first observe that:

$$\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) = \perp] \leq e^{\varepsilon} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') = \perp]$$

This is immediate since the indicator for a failure is a post-processing of the Laplace mechanism. Since x_1 can affect at most one oracle failure, T is 1-sensitive, and so publishing $\tilde{T} = T + \mathrm{Lap}\left(\frac{1}{\varepsilon}\right)$ satisfies ε-differential privacy since it is an invocation of the Laplace Mechanism defined in Section 2.1. (This can also be viewed as an instantiation of the "sub-sample and aggregate procedure of Nissim et al. (2007)).

We now proceed to the meat of the argument. To establish (ε, δ) differential privacy we must reason about the probability of arbitrary events $\Omega \subset \mathcal{M}$, rather than just individual outputs a. We want to show:

$$\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega] \leq e^{\varepsilon} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega] + \delta$$

We first fix some notation and define a number of events that we will need to reason about. Let:

- $\mathcal{P}_{split}^{S}$ be the uniform distribution over equal sized partitions of S that the datasets S_i are drawn from in line 1; i.e. $\mathcal{P}(S) \sim \mathcal{P}_{split}^{S}$, where $\mathcal{P}(S)$ is the partition of S into $\{S_i\}$.
- $\mathcal{A}$ denote $\mathcal{A}_{\mathcal{O}^*}$, our oracle-efficient algorithm when instantiated with a perfect oracle $\mathcal{O}^*$. i.e. $\mathcal{A}(S)$ is the ε-differentially private distribution that we ideally want to sample from.
- Z be the event that the Laplace noise z in step 10 of **PRSMA** has magnitude greater than $\frac{1}{\varepsilon}(\log(\frac{2}{\delta}))$.

- $\mathcal{F} = \left\{\{o_i\} : |\{o_i : o_i \neq \perp\}| > \frac{1}{\varepsilon}\right\}$. We will use $\mathbf{o}$ to denote a particular set $\{o_i\}$. Let $I^{\mathbf{o}}_{pass}$ be the set $\{i : o_i \neq \perp\}$. Given $\mathbf{o} \in \mathcal{F}$, let $|\mathbf{o}|$ denote $|I^{\mathbf{o}}_{pass}|$.

- E be the event that for all $i = 1 \dots K$: $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) = \perp] \geq \delta \Rightarrow o_i = \perp$.

- i^* denote the index i of the randomly chosen a_{it} in step 11 of **PRSMA**.

- Q be the event that the draw $\mathcal{P}(S) \sim \mathcal{P}^S_{split}$ is such that the probabilities of $\mathcal{A}$ outputting $a \in \Omega$ when run on any two $S_i, S_j \in \mathcal{P}(S)$ are within a multiplicative factor of 2. Lemma 4.3.10 formally defines Q and shows $\Pr[Q] \geq 1 - \delta$. Let S_Q denote the set of $\mathcal{P}(S)$ on which event Q holds.

We now bound the probabilities of several of these events. By the CDF of the Laplace distribution, we have $\Pr[Z] = \Pr\left[|z| > \frac{1}{\varepsilon}\log(1/\delta)\right] = e^{-\varepsilon \cdot \frac{1}{\varepsilon}\log(1/\delta)} = \delta$, and by a union bound:

$$\Pr[E] \geq 1 - K \cdot (1-\delta)^{(\log(K/\delta)/\delta)} \geq 1 - K \cdot e^{-\delta \cdot \log(K/\delta)/\delta} = 1 - \delta.$$

Let $\mathcal{L}$ be the event $Z^c \cap E$. By the above calculation and another union bound, $\Pr[\mathcal{L}] \geq 1 - 2\delta$. Our proof now proceeds via a sequence of lemmas. All missing proofs appear in Appendix A.2.3. We first show that Q occurs with high probability.

Lemma 4.3.10. *Let $\mathcal{P}(S) \sim \mathcal{P}^S_{split}$. Let $\mathcal{A} : \mathcal{X}^l \to \mathcal{M}$ be an $(\varepsilon', 0)$ differentially private algorithm, where: $\varepsilon' = \frac{1}{\sqrt{8\frac{n}{K}\log(2K/\delta)}}$. Fix $\Omega \subset M$, and let $q_\Omega(S_i) = \log \Pr[\mathcal{A}(S_i) \in \Omega]$. Define Q to be the event*

$$Q = \{\mathcal{P}(S) : \max_{i,j \in 1 \dots K} |q_\Omega(S_i) - q_\Omega(S_j)| \leq 2\}.$$

Then over the random draw of $\mathcal{P}(S) \sim \mathcal{P}^S_{split}$, $\Pr[Q] \geq 1 - \delta$.

The proof relies on the fact that $q_\Omega(\cdot)$ is ε-Lipschitz. This result is similar to Theorem 5.4 in Cummings et al. (2016), although in our case sampling without replacement induces dependence among the elements in S_i, and thus we can't appeal to standard concentration inequalities for independent random variables. Instead we prove that elements sampled

without replacement from a fixed set satisfy a type of negative dependence called *stochastic covering*, and are n/K-homogenous (supported on a set of size n/K), which are used to prove exponential concentration of Lipschitz functions in PEMANTLE and PERES (2014). We defer the details to the Appendix.

To establish the theorem we want to show that given an adjacent database S' differing only in the first element from S, that

$$\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \leq e^{\varepsilon^*} + \frac{\delta^*}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \tag{4.2}$$

Our analysis will proceed by expanding the numerator by first conditioning on $\mathcal{L}$, and then on particular realizations of the partition $\mathcal{P}(S)$, and on a fixed realization of $\mathbf{o} = \{o_i\}$. We can also restrict our attention to only summing over $\mathcal{P}(S) \in S_Q$, by showing that the terms corresponding to $\mathcal{P}(S) \in S_Q^c$ contribute at most an additive factor of 2δ to the final probability. We will also only sum over $\mathbf{o} \in \mathcal{F}$ since conditioned on Z^c, which is implied by $\mathcal{L}$, these are the only $\mathbf{o}$ such that $\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | \mathbf{o}] \neq 0$.

Lemma 4.3.11.

$$\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \leq \frac{\sum_{\mathcal{P}(S) \in S_Q, \boldsymbol{o} \in \mathcal{F}} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \boldsymbol{o}, \mathcal{L}] \Pr[\boldsymbol{o}, P(S) | \mathcal{L}] + 4\delta}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]}$$

The rest of the proof will consist of upper bounding the individual terms $\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S) | \mathcal{L}]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]}$. Lemma 4.3.12 is a tool used to prove Lemma 4.3.13, which upper bounds the numerator, and Lemma 4.3.14 lower bounds the denominator. The conclusion of the argument consists of manipulations to upper bound the ratio of these two bounds.

We first analyze the $\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}]$ term, for $\mathcal{P}(S) \in S_Q, \mathbf{o} \in \mathcal{F}$. Conditioned on Z^c, if $\mathbf{o} \in \mathcal{F}$, then **PRSMA** passes the test in step 10 and outputs a randomly chosen $a_{it} : i \in$

$I^{\mathbf{o}}_{pass}$. Fixing a sampled value $i^* \in I^{\mathbf{o}}_{pass}$, a_{i^*t} is distributed identically to $a \sim \mathcal{A}_{\mathcal{O}}(S_{i^*})|\mathcal{A}_{\mathcal{O}}(S_{i^*}) \neq \perp$, since after conditioning on S_{i^*} each a_{i^*t} is drawn iid and the event $o_i \neq \perp$ does not depend on the sampled values a_{it}. In other words, $\Pr[\mathcal{A}^{prsm}_{\mathcal{O}}(\mathcal{S}) \in \Omega | i^* = i, P(S), \mathbf{o}] = \Pr[\mathcal{A}_{\mathcal{O}}(S_{i^*}) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_{i^*}) \neq \perp]$, and so:

$$\Pr[\mathcal{A}^{prsm}_{\mathcal{O}}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] = \frac{1}{|\mathbf{o}|} \sum_{i \in I^{\mathbf{o}}_{pass}} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \tag{4.3}$$

Substituting in 4.3 we have:

$$\sum_{\mathcal{P}(S) \in S_Q, \mathbf{o} \in \mathcal{F}} \Pr[\mathcal{A}^{prsm}_{\mathcal{O}}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S) | \mathcal{L}] = \sum_{\mathcal{P}(S) \in S_Q, \mathbf{o} \in \mathcal{F}} \frac{1}{|\mathbf{o}|} \sum_{i \in I^{\mathbf{o}}_{pass}} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \Pr[\mathbf{o}, P(S) | \mathcal{L}] \tag{4.4}$$

We now use the fact that $\mathcal{P}(S) \in S_Q$, to show that none of the terms $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp]$ in the right hand side of Equation 4.3 individually represent a substantial fraction of the total probability mass. Conditioning on $\mathcal{L}$ ensures that if $o_i \neq \perp$ then $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) = \perp] \leq \delta$, which means $\mathcal{A}_{\mathcal{O}}(S_i)$ is δ-close in total variation distance to $\mathcal{A}(S_i)$. By Lemma 4.3.10, with high probability any $\mathcal{A}(S_i)$ is approximately equally likely to output an element in Ω, which allows us to bound the effect that changing a single data point (and hence a single S_i) can have on the probability of outputting an element in Ω.

Lemma 4.3.12. *Fix any $\boldsymbol{o}$, any $\mathcal{P}(S) \in S_Q$, and index $j \in I^{\boldsymbol{o}}_{pass}$, i.e. $o_j \neq \perp$. Then:*

$$\Pr[\mathcal{A}_{\mathcal{O}}(S_j) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_j) \neq \perp, \mathcal{L}] \leq \frac{e^2}{(1-\delta)^2} \frac{1}{|\boldsymbol{o}| - 1} \sum_{i \in I^{\boldsymbol{o}}_{pass}, i \neq j} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] + \frac{\delta e^2}{(1-\delta)}$$

Without loss of generality (up to renaming of partition elements), assume that the element on which S and S' differ falls into S_1. Now we break the summation over $\mathcal{O}$ into two pieces, depending on whether $o_1 = \perp$ or $o_1 \neq \perp$. We will use Lemma 4.3.12 to bound terms

involving $\Pr[\mathcal{A}_{\mathcal{O}}(S_1) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_1) \neq \perp, \mathcal{L}]$, since S_1 is the only partition where $S_i \neq S_i'$.

$$\sum_{P(S)\in S_Q} \Bigg(\sum_{\mathbf{o}\in\mathcal{F}} \Big(\frac{1}{|\mathbf{o}|} \sum_{i\in I^{\mathbf{o}}_{pass}} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \Big) \Pr[\mathbf{o}|P(S)] \Bigg) \Pr[P(S)] =$$

$$\sum_{P(S)\in S_Q} \Bigg(\sum_{\mathbf{o}\in\mathcal{F}: o_1 \neq \perp} \Big(\frac{1}{|\mathbf{o}|} \sum_{i\in I^{\mathbf{o}}_{pass}} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \Big) \Pr[\mathbf{o}|P(S)] +$$

$$\sum_{\mathbf{o}\in\mathcal{F}: o_1 = \perp} \Big(\frac{1}{|\mathbf{o}|} \sum_{i\in I^{\mathbf{o}}_{pass}} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \Big) \Pr[\mathbf{o}|P(S)) \Bigg) \Pr[P(S)]$$

Lemma 4.3.13.

$$\sum_{\mathcal{P}(S)\in S_Q, \boldsymbol{o}\in\mathcal{F}} \Pr[\mathcal{A}^{prsm}_{\mathcal{O}}(\mathcal{S}) \in \Omega | P(S), \boldsymbol{o}, \mathcal{L}] \Pr[\boldsymbol{o}, P(S) | \mathcal{L}] \leq$$

$$(1 + \frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon} - 1)}) \sum_{P(S)\in S_Q} \Bigg(\sum_{\boldsymbol{o}\in\mathcal{F}} \Big(\frac{1}{|\boldsymbol{o}|} \sum_{i\in I^{\mathbf{o}}_{pass}, i\neq 1} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \Big) \Pr[\boldsymbol{o}|P(S)] \Bigg)$$
$$\Pr[P(S)] + \frac{\varepsilon\delta e^2}{1-\delta} \quad (4.5)$$

We now condition on a fixed partition $P(S')$ in the denominator as well. Given a fixed partition $P(S)$ of S, define the adjacent partition $\mathcal{P}(S') \sim \mathcal{P}(S)$ as the partition of an adjacent database S' such that for all $i \neq 1$, $S_i = S_i'$, and S_1, S_1' differ only in $x_1 \neq x_1'$, where x_1 is the differing element between S, S'. Let S_Q' be the set of $\mathcal{P}(S')$ adjacent to $\mathcal{P}(S) \in S_Q$, i.e. $S_Q' = \{\mathcal{P}(S') : \exists\ \mathcal{P}(S) \in S_Q, \mathcal{P}(S') \sim \mathcal{P}(S)\}$. We now lower bound the denominator in Lemma 4.3.11, which follows by conditioning on $\mathbf{o}, \mathcal{P}(S')$, and then dropping some (non-negative) terms.

Lemma 4.3.14.

$$\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega] \geq \sum_{P(S') \in S'_Q} \Big(\sum_{\boldsymbol{o} \in \mathcal{F}} \Big(\frac{1}{|\boldsymbol{o}|} \sum_{i \in I^{\boldsymbol{o}}_{pass}, i \neq 1} \Pr[\mathcal{A}_{\mathcal{O}}(S'_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S'_i) \neq \perp] \Big) \Pr[\boldsymbol{o} | P(S')] \Big) \Pr[P(S')]$$

Thus by Lemmas 4.3.13 and 4.3.14,

$$\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S) | \mathcal{L}]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \leq$$

$$\frac{(1 + \frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)}) \sum_{P(S) \in S_Q} \Big(\sum_{\mathbf{o} \in \mathcal{F}} \Big(\frac{1}{|\mathbf{o}|} \sum_{i \in I^{\mathbf{o}}_{pass}, i \neq 1} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \Pr[\mathbf{o} | P(S)] \Big) \Pr[P(S)] \Big)}{\sum_{P(S') \in S'_Q} \Big(\sum_{\mathbf{o} \in \mathcal{F}} \Big(\frac{1}{|\mathbf{o}|} \sum_{i \in I^{\mathbf{o}}_{pass}} \Pr[\mathcal{A}_{\mathcal{O}}(S'_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S'_i) \neq \perp] \Big) \Pr[\mathbf{o} | P(S')] \Big) \Pr[P(S')]} +$$

$$\frac{\frac{\varepsilon \delta e^2}{1-\delta}}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \quad (4.6)$$

For all $\mathcal{P}(S), \mathcal{P}(S'), \Pr[\mathcal{P}(S)] = \Pr[\mathcal{P}(S')]$, and we can bound the ratio of the summations over S_Q, S'_Q by the supremum of the ratio. Hence (4.6) $\leq$

$$\sup_{P(S) \sim P(S')} \frac{(1 + \frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)}) \sum_{\mathbf{o} \in \mathcal{F}} \Big(\frac{1}{|\mathbf{o}|} \sum_{i \in I^{\mathbf{o}}_{pass}, i \neq 1} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \Pr[\mathbf{o} | P(S)] \Big)}{\sum_{\mathbf{o} \in \mathcal{F}} \Big(\frac{1}{|\mathbf{o}|} \sum_{i \in I^{\mathbf{o}}_{pass}} \Pr[\mathcal{A}_{\mathcal{O}}(S'_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S'_i) \neq \perp] \Pr[\mathbf{o} | P(S')] \Big)} +$$

$$\frac{\frac{\varepsilon \delta e^2}{1-\delta}}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \quad (4.7)$$

Now since for all $i \neq 1, S_i = S'_i$, if we could control the ratio $\Pr[\mathbf{o}|\mathcal{P}(S)]/\Pr[\mathbf{o}|\mathcal{P}(S')]$ we would be done. But this ratio could potentially be unbounded, as $\Pr[\mathbf{o}_1|P(S)]$ could be nonzero, and the substitution of x'_1 for x_1 could force failure on the first partition S'_1, and so $\Pr[\mathbf{o}_1|P(S')] = 0$.

Given $\mathbf{o}$ let $\mathbf{o}_{-1}$ denote $\{o_2, \dots o_K\}$. The remainder of the argument circumvents this obstacle by decomposing the outer summation over $\mathbf{o} \in \mathcal{F}$ into a summation over the indicators of all but the first failure event ($\mathbf{o}_{-1}$), and integrating out the probability of the first failure event (o_1) from the joint probability $\Pr[\mathbf{o}|P(S)]$. This trick will be applied in the numerator and denominator, with a slight difference corresponding to an upper and a lower bound respectively. See the end of Section **C** of the Appendix for details. Following the chain of inequalities, we finally obtain:

$$\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S)|\mathcal{L}]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \leq (1 + \frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon} - 1)}) \frac{1}{1-\varepsilon} + \frac{\frac{\varepsilon\delta e^2}{1-\delta}}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]},$$

which substituting into Lemma 4.3.11 gives:

$$\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega] \leq (1 + \frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon} - 1)}) \frac{1}{1-\varepsilon} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega] + 4\delta + \frac{\varepsilon\delta e^2}{1-\delta}$$

For $\varepsilon, \delta \leq 1/2$, $(1 + \frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)}) \frac{1}{1-\varepsilon} \leq e^{8e^2\varepsilon + \varepsilon + \varepsilon^2}$, which establishes that **PRSMA** is $(8e^2\varepsilon + \varepsilon + \varepsilon^2, 4\delta + \frac{\varepsilon\delta e^2}{1-\delta})$ differentially private. Setting $\varepsilon = \varepsilon^*, \delta = \delta^*$ completes the proof.

□

We now turn to **PRSMA**'s accuracy guarantees. Note that when **PRSMA** starts with an algorithm $\mathcal{A}_{\mathcal{O}^*}$ instantiated with a perfect oracle $\mathcal{O}^*$, it with high probability outputs the result of running $\mathcal{A}_{\mathcal{O}^*}$ on a subsampled dataset S_i of size $n/K \approx \varepsilon n$, with privacy parameter $\varepsilon' = \frac{1}{\sqrt{8\frac{n}{K}\log(2K/\delta)}}$. In general, therefore, the accuracy guarantees of **PRSMA** depend on how robust the guarantees of $\mathcal{A}$ are to subsampling, which is typical of "Subsample and Aggregate" approaches, and also to its specific privacy-accuracy tradeoff. Learning algorithms are robust to sub-sampling however: below we derive an accuracy theorem for **PRSMA** when instantiated with our oracle-efficient RSPM algorithm.

Theorem 4.3.15. *Let $\mathcal{Q}$ a class of statistical queries with a separator set of size m. Let $\mathcal{A}_{\mathcal{O}^*}$ denote the RSPM algorithm with access to $\mathcal{O}^*$, a perfect weighted optimization oracle for $\mathcal{Q}$. Then*

PRSMA *instantiated with $\mathcal{A}_{\mathcal{O}^*}$, run on a dataset S of size n, with input parameters ε and δ is an (α, β)-minimizer for any $\beta > \delta$ and*

$$\alpha \leq \tilde{O}\left(\frac{m^2 \log\left(\frac{m}{\beta-\delta}\right)\log(1/\delta) + \sqrt{\log(1/\delta)\log\left(\frac{|\mathcal{Q}|}{\beta-\delta}\right)}}{\sqrt{n\varepsilon}}\right),$$

where the $\tilde{O}$ hides logarithmic factors in $\frac{1}{\varepsilon}, \log(\frac{1}{\delta})$.

Proof. With probability at least $1-\delta$, **PRSMA** outputs the result of RSPM run on an n/K fraction of the dataset, with privacy parameter $\varepsilon' = \frac{1}{\sqrt{8\frac{n}{K}\log(2K/\delta)}}$. We will condition on this event for the remainder of the proof, which occurs except with probability δ.

Let q^* denote the true minimizer on S. Let S_K denote the random subsample, and let q_K denote the true minimizer on S_K. By Theorem 4.3.2, we know that for any $\eta > 0$, with probability $1-\eta$, the error on S_K is bounded as follows:

$$\hat{q}(S_K) - q_K(S_K) \leq \frac{2m^2 \log(m/\eta)}{\varepsilon' \frac{n}{K}}$$

We next bound $\max_{q\in\mathcal{Q}} q(S_K) - q(S)$, the maximum difference between the value that any query takes on S_K compared to the value that it takes on S. By a Chernoff bound for subsampled random variables (see e.g. Theorem 1.2 of Bardenet and Maillard (2013)), for any $q \in \mathcal{Q}, t > 0$,

$$\Pr\left[q(S_K) - q(S) \geq t\right] \leq \exp\left(-2\frac{n}{K}t^2\right).$$

By a union bound over $\mathcal{Q}$, this means that with probability $1-\eta$,

$$\max_{q\in\mathcal{Q}} q(S_K) - q(S) \leq \sqrt{\frac{K}{2n}\log\left(\frac{2|\mathcal{Q}|}{\eta}\right)}$$

We now have all the ingredients to complete the bound:

$$\begin{aligned}\hat{q}(S) - q^*(S) &= [\hat{q}(S) - \hat{q}(S_K)] + [\hat{q}(S_K) - q^*(S_K)] + [q^*(S_K) - q^*(S)] \\ &\leq 2\max_{q\in\mathcal{Q}} |q(S_K) - q(S)| + \hat{q}(S_K) - q^*(S_K) \\ &= 2\max_{q\in\mathcal{Q}} |q(S_K) - q(S)| + \hat{q}(S_K) - q_K(S_K) + q_K(S_K) - q^*(S_K) \\ &\leq 2\max_{q\in\mathcal{Q}} |q(S_K) - q(S)| + \hat{q}(S_K) - q_K(S_K).\end{aligned}$$

By a union bound and the results above, we know that the righthand side is less than

$$2\sqrt{\frac{K}{2n}\log(\frac{2|\mathcal{Q}|}{\eta})} + \frac{2m^2\log(m/\eta)}{\varepsilon'\frac{n}{K}},$$

with probability at least $1-\eta$. Substituting $\eta = \beta - \delta$, $\varepsilon' = \frac{1}{\sqrt{8\frac{n}{K}\log(2K/\delta)}}$, $K = O(\frac{1}{\varepsilon}(1+\log(2/\delta)))$ gives the desired result. □

We remark that we can convert this (α,β)-accuracy bound into a bound on the expected error using the same technique we used to compute the expected error of RSPM. The expected error of **PRSMA** with the above inputs is $\tilde{O}(\frac{2m^2(\log m+1)}{\sqrt{n\varepsilon}} + \frac{\sqrt{(\log|\mathcal{Q}|+1)}}{\sqrt{n\varepsilon}})$.

4.4. OracleQuery: Oracle-Efficient Private Synthetic Data Generation

We now apply the oracle-efficient optimization methods we have developed to the problem of generating *private synthetic data*. In particular, given a private dataset S and a query class $\mathcal{Q}$, we would like to compute a synthetic dataset $\hat{S}$ subject to differential privacy such that the error $\max_{q\in\mathcal{Q}} |q(\hat{S}) - q(S)|$ is bounded by some target parameter α. We provide a general algorithmic framework called OracleQuery for designing oracle-efficient algorithms. The crucial property of the query class we rely on to obtain oracle efficiency is *dual separability*, which requires both the query class and its dual class have separator sets (Definition 4.2.1). Informally, the dual of a query class $\mathcal{Q}$ is the query class $\mathcal{Q}_{\text{dual}}$ that results from swapping the role of the functions $q \in \mathcal{Q}$ and the data elements $x \in \mathcal{X}$. More formally:

Definition 4.4.1 (Dual class and dual separability)**.** *Fix a class of queries $\mathcal{Q}$. For every element x in $\mathcal{X}$, let $h_x \colon \mathcal{Q} \to \{0,1\}$ be defined such that $h_x(q) = q(x)$. The* dual class $\mathcal{Q}_{\text{dual}}$ *of $\mathcal{Q}$ is the set of all such functions defined by elements in $\mathcal{X}$:*

$$\mathcal{Q}_{\text{dual}} = \{h_x \mid x \in \mathcal{X}\}.$$

We say that the class $\mathcal{Q}$ is (m_1, m_2)-dually separable if there exists a separator set of size m_1 for $\mathcal{Q}$, and there exists a separator set of size m_2 for $\mathcal{Q}_{\text{dual}}$.

As we will show (see Appendix A.2.1), many widely studied query classes, including discrete halfspaces, conjunctions, disjunctions, and parities are dually separable, often with $m_1 = m_2 = d$ (in fact, many of these classes are *self-dual*, meaning $\mathcal{Q} = \mathcal{Q}_{\text{dual}}$). For any $q \in \mathcal{Q}$, define its negation $\neg q$ to be $\neg q(x) = 1 - q(x)$. Let $\neg\mathcal{Q} = \{\neg q \mid q \in \mathcal{Q}\}$ be the *negation of* $\mathcal{Q}$. It will simplify several aspects of our exposition to deal with classes that are closed under negation. For any class $\mathcal{Q}$, define $\overline{\mathcal{Q}} = \mathcal{Q} \cup \neg\mathcal{Q}$ to be the closure of $\mathcal{Q}$ under negation. Note that whenever we have a weighted minimization oracle for $\mathcal{Q}$, we have one for $\neg\mathcal{Q}$ as well — simply by negating the weights. Further, if U is a separator set for $\mathcal{Q}$, it is also a separator set for $\neg\mathcal{Q}$. This implies that we also have oracle efficient learners for $\overline{\mathcal{Q}}$, since we can separately learn over $\mathcal{Q}$ and $\neg\mathcal{Q}$, and then privately take the minimum value query that results from the two procedures (using e.g. report-noisy-min Dwork and Roth (2014b)).

Before we give our algorithm and analysis, we state several consequences of our main theorem (that follow from instantiating it with different oracle-efficient learners).

Theorem 4.4.2. *Let $\mathcal{Q}$ be an (m_1, m_2)-dually separable query class. Then given access to a weighted minimization oracle $\mathcal{O}$ over the class $\mathcal{Q}_{\text{dual}}$ and a differentially private weighted minimization algorithm $\mathcal{O}_{\varepsilon_0,\delta_0}$ for the class $\mathcal{Q}$ (with appropriately chosen privacy parameters ε_0 and δ_0), the algorithm* OracleQuery *is oracle-efficient, (ε,δ)-differentially private, and (α,β)-accurate with α depending on the instantiation of $\mathcal{O}_{\varepsilon_0,\delta_0}$. If $\mathcal{O}_{\varepsilon_0,\delta_0}$ is robustly differentially private, then so is* OracleQuery.

1. *If $\mathcal{O}_{\varepsilon_0,\delta_0}$ is instantiated with the Gaussian RSPM algorithm, then*

$$\alpha \leq \tilde{O}\left(\frac{m_1^{3/2} m_2^{3/4} \sqrt{\log(m_1/\beta)\log|\mathcal{X}|\log(1/\delta)}}{n\varepsilon}\right)^{1/2}$$

In this case, OracleQuery *is oracle equivalent to a differentially private algorithm, but is not robustly differentially private.*

2. *If $\mathcal{O}_{\varepsilon_0,\delta_0}$ is instantiated with the **PRSMA** algorithm (using the Laplace RSPM as $\mathcal{A}_{\mathcal{O}^*}$), then*

$$\alpha \leq \tilde{O}\left(\frac{(m_1^{4/3} + \log^{1/3}(|\mathcal{Q}|))m_2^{1/4}\log^{1/6}(|\mathcal{X}|)}{(n\varepsilon)^{1/3}}\right) \cdot \text{polylog}\left(\frac{1}{\beta - \delta}\right)$$

as long as $\beta > \delta$. In this case, OracleQuery *is robustly differentially private.*

3. *If $\mathcal{O}_{\varepsilon_0,\delta_0}$ is an (α_0, β_0)-accurate differentially private oracle with $\alpha_0 = O(\log(|\mathcal{Q}|/(\varepsilon_0 n)))$, then*

$$\alpha \leq \tilde{O}\left(\frac{m_2^{3/4}\sqrt{\log|\mathcal{X}|\log(1/\delta)}\log(|\mathcal{Q}|/\beta)}{n\varepsilon}\right)^{1/2}$$

In this case, OracleQuery *is robustly differentially private.*

where the $\tilde{O}$ hides logarithmic factors in $\frac{1}{\delta}, \frac{1}{\beta}, m_1, m_2, n$ and $\log(|\mathcal{X}|)$.

A couple of remarks are in order.

Remark 4.4.3. *The first two bounds quoted in Theorem 4.4.2 result from plugging in constructions of oracle-efficient differentially private learners that we gave in Section 4.3. These constructions start with a* non-private *optimization oracle. The third bound quoted in Theorem 4.4.2 assumes the existence of a* differentially private *oracle with error bounds comparable to the (inefficient) exponential mechanism based learner of Kasiviswanathan et al. (2011). We don't know if such oracles can be constructed from non-private (exact) optimization oracles. But this bound is analgous to the bounds given in the non-private oracle-efficient learning literature. This literature gives constructions assuming the existence of perfect learning oracles, but in practice, these oracles are instantiated with heuristics like regression or support vector machines, which*

exactly optimize some convex surrogate loss function. This is often reasonable, because although these heuristics don't have strong worst-case guarantees, they often perform very well in practice. The same exercise makes sense for private problems: we can use a differentially private convex minimization algorithm to optimize a surrogate loss function (e.g. Chaudhuri et al. (2011); Bassily et al. (2014)), and hope that it does a good job minimizing classification error in practice. It no longer makes sense to assume that the heuristic exactly solves the learning problem (since this is impossible subject to differential privacy) — instead, the analogous assumption is that it does as well as the best inefficient private learner.

Remark 4.4.4. *It is useful to compare the bounds we obtain to the best bounds that can be obtained with inefficient algorithms. To be concrete, consider the class of boolean conjunctions defined over the boolean hypercube $\mathcal{X} = \{0,1\}^d$ (see Appendix A.2.1), which are dually-separable with $m_1 = m_2 = d$. The best (inefficient) bounds for constructing synthetic data useful for conjunctions Hardt and Rothblum (2010); Gupta et al. (2012) obtain error: $\alpha = O\left(\frac{\sqrt{\log|\mathcal{Q}|}(\log|\mathcal{X}|)^{1/4}}{\sqrt{\varepsilon n}}\right)$. In the case of boolean conjunctions, $\log|\mathcal{X}| = \log|\mathcal{Q}| = d$, and so this bound becomes: $\alpha = O\left(\frac{d^{3/4}}{\sqrt{\varepsilon n}}\right)$. In contrast, the three oracle efficient bounds given in Theorem 4.4.2, when instantiated for boolean conjunctions are:*

1. $\alpha = O\left(\frac{d^{11/8}}{\sqrt{\varepsilon n}}\right)$,
2. $\alpha = O\left(\frac{d^{7/4}}{(\varepsilon n)^{1/3}}\right)$, *and*
3. $\alpha = O\left(\frac{d^{9/8}}{\sqrt{\varepsilon n}}\right)$

respectively. Therefore the costs in terms of error that we pay, in exchange for oracle efficiency are $d^{5/8}$, $\frac{d}{(\varepsilon n)^{1/6}}$, and $d^{3/8}$ respectively.

We now give a brief overview of our construction before diving into the technical details.

Proof overview: We present our solution in three main steps.

1. We first revisit the formulation by Hsu et al. (2013b) that views the synthetic data generation problem as a zero-sum game between a *Data player* and a *Query player*.

We leverage the fact that at any approximate equilibrium, the data player's mixed strategy (over $\mathcal{X}$) represents a good synthetic dataset S' with respect to $\mathcal{Q}$.

2. Using the seminal result of Freund and Schapire (1996a), we will compute the equilibrium for the zero-sum game by simulating *no-regret dynamics* between the two players: in rounds, the Data player plays according to an oracle-efficient online learning algorithm due to Syrgkanis et al. (2016), and the Query player best responds to the Data player by using a differentially private oracle efficient optimization algorithm. At the end of the dynamics, the average play of the Data player is an approximate minimax strategy for the game, and hence a good synthetic dataset.

3. We instantiate the private best response procedure of the Query player using different oracle-efficient methods, which we have derived in this paper, each of which gives different accuracy guarantees. Finally, we apply our result to several query classes of interest.

4.4.1. The Query Release Game

The query release game defined in Hsu et al. (2013b) involves a *Data player* and *Query player*. The data player has action set equal to the data universe $\mathcal{X}$ (or equivalently the dual class $\mathcal{Q}_{\text{dual}}$), while the query player has action set equal to the query class $\overline{\mathcal{Q}}$. Given a pair of actions $x \in \mathcal{X}$ and $q \in \overline{\mathcal{Q}}$, the payoff is defined to be:

$$A(x, q) = q(S) - q(x),$$

where S is the input private dataset. In the zero-sum game, the Data player will try minimize the payoff and the Query player will try to maximize the payoff. To play the game, each player chooses a *mixed strategy*, which is defined by a probability distribution over their action set. Let $\Delta(\mathcal{X})$ and $\Delta(\overline{\mathcal{Q}})$ denote the sets of *mixed strategies* of the Data player and Query player respectively. To simplify notation, we will write $A(\hat{S}, \cdot) = \mathbb{E}_{x \sim \hat{S}}[A(x, \cdot)]$ and $A(\cdot, W) = \mathbb{E}_{q \sim W}[A(\cdot, q)]$ for any $\hat{S} \in \Delta(\mathcal{X})$ and $W \in \Delta(\overline{\mathcal{Q}})$. By von Neumann's minimax

theorem, there exists a value V such that

$$V = \min_{\hat{S} \in \Delta(\mathcal{X})} \max_{q \in \overline{\mathcal{Q}}} A(\hat{S}, W) = \max_{W \in \Delta(\overline{\mathcal{Q}})} \min_{x \in \mathcal{X}} A(x, W)$$

If both players are playing strategies that can guarantee a payoff value close to V, then we say that the pair of strategies form an approximate equilibrium.

Definition 4.4.5 (Approximate Equilibrium)**.** *For any $\alpha > 0$, a pair of strategies $\hat{S} \in \Delta(\mathcal{X})$ and $W \in \Delta(\overline{\mathcal{Q}})$ form an* α-approximate minimax equilibrium *if*

$$\max_{q \in \overline{\mathcal{Q}}} A(\hat{S}, q) \leq V + \alpha \qquad \text{and} \qquad \min_{x \in \mathcal{X}} A(W, x) \geq V - \alpha.$$

Hsu et al. Hsu et al. (2013b) show that the query release game has value $V = 0$ and that at any approximate equilibrium, the mixed strategy of the Data player provides accurate answers for all queries in $\overline{\mathcal{Q}}$.

Lemma 4.4.6 (Accuracy at equilibrium Hsu et al. (2013b))**.** *Let $(\hat{S}, W)$ be an α-approximate equilibrium of the query release game. Then for any $q \in \overline{\mathcal{Q}}$, $|q(S) - q(\hat{S})| \leq \alpha$.*

Therefore, the dataset represented by the distribution $\hat{S}$ (or that could be obtained by sampling from $\hat{S}$) is exactly the synthetic dataset we would like to compute, and hence the problem of privately computing synthetic data is reduced to the problem of differentially private equilibrium computation in the query release game.

4.4.2. Solving the Game with No-Regret Dynamics

To privately compute an approximate equilibrium of the game, we will simulate the following *no-regret dynamics* between the Data Player and the Query Player in rounds: In each round t, the Data player plays a distribution S^t according to a *no-regret* learning algorithm, and the Query player plays an approximate best-response to S^t. The following classical theorem of Freund and Schapire Freund and Schapire (1996a) (instantiated in our

setting) shows that the average play of both players in this dynamic forms an approximate equilibrium.

Theorem 4.4.7 (Freund and Schapire (1996a)). *Let $S^1, S^2, \ldots, S^T \in \Delta(\mathcal{X})$ be a sequence of distributions played by the Data Player, and let $q^1, q^2, \ldots, q^T \in \overline{\mathcal{Q}}$ be the Query player's sequence of approximate best-responses against these distributions. Suppose that the* regret *of the two players satisfy:*

$$\mathrm{Reg}_D(T) = \sum_{t=1}^{T} A(S^t, q^t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} A(x, q^t) \leq \gamma_D T$$
$$\mathrm{Reg}_Q(T) = \max_{q \in \overline{\mathcal{Q}}} \sum_{t=1}^{T} A(S^t, q) - \sum_{t=1}^{T} A(S^t, q^t) \leq \gamma_Q T.$$

Let $\overline{S}$ be uniform mixture of the distributions $\{S^1, \ldots, S^T\}$ and $\overline{W}$ be the uniform distribution over $\{q^1, \ldots, q^T\}$. Then $(\overline{S}, \overline{W})$ is a $(\gamma_D + \gamma_Q)$-approximate minimax equilibrium of the game.

Now we will detail the no-regret algorithm for the Data player and the best-response method for the Query player, and provide the regret bounds γ_D and γ_Q.

No-Regret Algorithm for the Data Player. We start with the observation that the regret of the Data player is independent of the private data S, because

$$\mathrm{Reg}_D(T) = \sum_{t=1}^{T} \left(q^t(S) - q^t(S^t) \right) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} \left(q^t(S) - q^t(x) \right) = \sum_{t=1}^{T} \neg q^t(S^t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} \neg q^t(x).$$

Therefore, it suffices to minimize regret with respect to the sequence of loss functions $\{\neg q^t(\cdot)\}$, while ignoring the private dataset S. We crucially rely on the fact that each q^t is computed by the Query player subject to differential privacy, and so the Data player's learning algorithm need not be differentially private: differential privacy for the overall procedure will follow from the post-processing guarantee of differential privacy. In particular, we will run an oracle-efficient algorithm Context-FTPL due to Syrgkanis et al. (2016), which is a variant of the "Follow-the-Perturbed-Leader" of Kalai and Vempala (2005) algorithm that performs perturbations using a separator set. We state its regret guarantee

below. Because Context-FTPL need not be differentially private, it can be instantiated with an arbitrary heuristic oracle, that need not be either differentially private or certifiable.

Context-FTPL $(\mathcal{Q}_{\text{dual}}, \mu)$ **Algorithm Syrgkanis et al. (2016)**
Given: parameter μ, hypothesis class $\mathcal{Q}_{\text{dual}}$ (or equivalently $\mathcal{X}$), separator set $U \subset \mathcal{Q}$ for $\mathcal{Q}_{\text{dual}}$, weighted optimization oracle $\mathcal{O}$ for $\mathcal{Q}_{\text{dual}}$
Input: A sequence of queries $\{q^1, \ldots, q^T\}$ selected by the Query player.

1: **for** $t = 1 \ldots T$ **do**
2: Data player plays the distribution S^t such that each draw x generated as follows:
3: Draw a sequence (s, η_s), for $s \in U$, where $\eta_s \sim \text{Lap}(\mu)$
4: Let $x = \text{argmin}_{x \in \mathcal{X}} \sum_{\tau=1}^{t-1} h_x(\neg q^\tau) + \sum_{s \in S} \eta_s h_x(s)$ ▷ Use non-private oracle $\mathcal{O}$

Theorem 4.4.8 (Follows from Syrgkanis et al. (2016))**.** *Suppose that U is a separator set for $\mathcal{Q}_{\text{dual}}$ of cardinality m_2. Then the Data player running Context-FTPL $(\mathcal{X}, \mu)$ with appropriately chosen μ has regret:*

$$\text{Reg}_D(T) \leq O(m_2^{3/4} \sqrt{T \log |\mathcal{X}|})$$

Note that the algorithm Context-FTPL only provides sample access to each distribution S^t, but each draw from S^t can be computed using a single call to the oracle $\mathcal{O}$.

Approximate Best Response by the Query Player. At each round t, after the Data player chooses S^t, the Query player needs to approximately solve the following best-response problem:

$$\underset{q \in \overline{\mathcal{Q}}}{\text{argmax}} A(S^t, q) = \underset{q \in \overline{\mathcal{Q}}}{\text{argmax}} \left(q(S) - q(S^t) \right)$$

Unlike the problem faced by the Data player, this optimization problem directly depends on the private data S, so the best response needs to be computed privately. Since we only have sample access to the distribution S^t, the Query player will first draw N random examples from the distribution S^t, and we will the empirical distribution $\hat{S}^t$ over the sample as a proxy for S^t. Recall that $\overline{\mathcal{Q}} = \mathcal{Q} \bigcup \neg \mathcal{Q}$, so we will first approximately and differentially privately solve both of the following two problems separately:

$$\underset{q \in \mathcal{Q}}{\text{argmax}} \left(q(S) - q(\hat{S}^t) \right) \qquad \text{and} \qquad \underset{q \in \neg \mathcal{Q}}{\text{argmax}} \left(q(S) - q(\hat{S}^t) \right) \tag{4.8}$$

Note that the two problems are equivalent to the following problems respectively:

$$\operatorname*{argmin}_{q\in\mathcal{Q}}\left(q(\hat{S}^t)-q(S)\right) \qquad \text{and} \qquad \operatorname*{argmin}_{q\in\mathcal{Q}}\left(q(S)-q(\hat{S}^t)\right), \tag{4.9}$$

both of which are weighted optimization problems:

$$\operatorname*{argmin}_{q\in\mathcal{Q}} \frac{1}{n}\sum_{x_i\in S} w_i q(x_i) + \frac{1}{N}\sum_{x'_j\in\hat{S}^t} w'_j q(x'_j) \tag{4.10}$$

with weights w_i, w'_j taking values in $\{1,-1\}$. We will rely on a private weighted optimization algorithm $\mathcal{O}_{\varepsilon_0,\delta_0}$ to compute two solutions q_1^t and q_2^t for the two problems in Equation (4.9) respectively. Finally, the Query player privately selects one of the queries using *report noisy max*—i.e. it first perturb the values of $A(\hat{S}^t, q_1^t)$ and $A(\hat{S}^t, q_2^t)$ with Laplace noise, and then select the query with higher noisy value. By bounding the errors from the sampling of $\hat{S}^t$, the private optimization oracle $\mathcal{O}_{\varepsilon_0,\delta_0}$, and report noisy max, we can derive the following regret guarantee for the Query player.

Private Best-Response (PBR)
Given: privacy parameters (ε_0,δ_0), accuracy parameters (α_0,β_0), a private weighted optimization algorithm $\mathcal{O}_{\varepsilon_0,\delta_0}$.
Input: A private dataset S and the Data player's sequence of distributions $\{S^1,\dots,S^T\}$.

1: **for** $t = 1\dots T$ **do**
2: Query player plays a query q^t as follows:
3: Draw N samples $x'_1,\dots x'_N \sim_{i.i.d.} S^t$ with $N = \frac{2\log(2|\mathcal{Q}|/\beta_0)}{\alpha_0^2}$
4: Form the weighted dataset $WD^1 = \{(x_i, \frac{1}{n})\}_{x_i\in S} \cup \{x_j, \frac{-1}{N}\}_{j=1\dots N}$
5: Form the weighted dataset $WD^2 = \{(x_i, \frac{-1}{n})\}_{x_i\in S} \cup \{x_j, \frac{1}{N}\}_{j=1\dots N}$
6: Let $q_1^t = \mathcal{O}_{\varepsilon_0,\delta_0}(WD^1)$ and $q_2^t = \neg\left(\mathcal{O}_{\varepsilon_0,\delta_0}(WD^2)\right)$
7: Perturb payoffs: $\tilde{A}_1 = A(\hat{S}^t, q_1^t) + \mathrm{Lap}\,(1/(\varepsilon_0 n))$ and $\tilde{A}_2 = A(\hat{S}^t, q_2^t) + \mathrm{Lap}\,(1/(\varepsilon_0 n))$
8: **If** $\tilde{A}_1 > \tilde{A}_2$ **then** $q^t = q_1^t$ **else** $q^t = q_2^t$

Lemma 4.4.9. *Suppose that the oracle $\mathcal{O}_{\varepsilon_0,\delta_0}$ succeeds in solving all problems it is presented with up to error at most α_0 except with probability β_0. Then with probability at least $1-3\beta_0 T$, the Query player has regret:*

$$\mathrm{Reg}_Q(T) \le T\, O\left(\alpha_0 + \frac{\log(1/\beta_0)}{n\varepsilon_0}\right).$$

Proof. There are three potential sources of error at each round t. The first is the error introduced by solving our optimization problem over the proxy distribution $\hat{S}^t$ instead of S^t. By applying a Chernoff bound and a union bound over all queries in $\mathcal{Q}$, we have with probability $1-\beta_0$ that,

$$\text{for all } q \in \overline{\mathcal{Q}}, \qquad \left|q(S^t) - q(\hat{S}^t)\right| \leq \sqrt{\frac{2\log(2|\mathcal{Q}|/\beta_0)}{N}} \leq \alpha_0. \tag{4.11}$$

Next is the error introduced by the oracle. By our assumption on the oracle $\mathcal{O}_{\varepsilon_0,\delta_0}$, we have except with probability $2\beta_0$ that

$$\left(q_1^t(S) - q_1^t(\hat{S}^t)\right) \geq \max_{q\in\mathcal{Q}}\left(q(S) - q(\hat{S}^t)\right) - \alpha_0 \qquad \text{and} \qquad \left(q_2^t(S) - q_2^t(\hat{S}^t)\right) \geq \max_{q\in\neg\mathcal{Q}}\left(q(S) - q(\hat{S}^t)\right) - \alpha_0$$

The two inequalities together imply that

$$\max_{q\in\{q_1^t, q_2^t\}}\left(q(S) - q(\hat{S}^t)\right) \geq \max_{q\in\neg\mathcal{Q}}\left(q(S) - q(\hat{S}^t)\right) - \alpha_0 \tag{4.12}$$

Finally, the Laplace noise used to privately select the best query amongst q_1^t and q_2^t introduces additional error. But by the accuracy guarantee of report noisy max Dwork and Roth (2014b) (which follows from the CDF of the Laplace distribution and a union bound over two samples from it) we know that with probability $1-\beta_0$,

$$\left(q^t(S) - q^t(\hat{S}^t)\right) \geq \max_{q\in\{q_1^t, q_2^t\}}\left(q(S) - q(\hat{S}^t)\right) - \frac{2\log(2/\beta_0)}{n\varepsilon_0} \tag{4.13}$$

Combining Equations (4.11) to (4.13) and applying a union bound, we have the following per-round guarantee: except with probability $3\beta_0$,

$$\left(q^t(S) - q^t(S^t)\right) \geq \max_{q\in\overline{\mathcal{Q}}}\left(q(S) - q(S^t)\right) - O\left(\alpha_0 + \frac{\log(1/\beta_0)}{n\varepsilon_0}\right).$$

Finally, taking a union bound over all T steps recovers the stated regret bound. □

4.4.3. *The Full Algorithm:* OracleQuery

Our main algorithm OracleQuery first simulates the no-regret dynamics described above, and then constructs a synthetic dataset from the average distribution $\overline{S} = \frac{1}{T}\sum_{t\in[T]} S^t$ played by the Data player. Since we only have sampling access to each S^t, we will approximate $\overline{S}$ by the empirical distribution of a set of independent samples drawn from $\overline{S}$. As we show below, the sampling error will be on the same order as the regret as long as we take roughly $\log|\mathcal{Q}|/\alpha^2$ samples.

Lemma 4.4.10. *Suppose that $\overline{S}$ is an η-approximate minimax strategy for the query release game. Let $\{x'_1,\ldots,x'_N\}$ be a set of $N = \frac{2\log(2|\mathcal{Q}|/\beta)}{\alpha_0^2}$ samples drawn i.i.d. from $\overline{S}$, and $\hat{S}$ be the empirical distributions over the drawn samples. Then with probability $1-\beta$, $\hat{S}$ is an $(\eta+\alpha_0)$-approximate minimax strategy.*

Proof. By the definition of an η-approximate minimax strategy, we have

$$\max_{q\in\overline{\mathcal{Q}}} A(\overline{S}, q) \le V + \eta = \eta.$$

By applying the Chernoff bound, we know that except with probability $\beta/|\mathcal{Q}|$, the following holds for each $q \in \mathcal{Q}$:

$$\left|A(\overline{S}, q) - A(\hat{S}, q)\right| \le \sqrt{\frac{2\log(2|\mathcal{Q}|/\beta)}{N}} = \alpha$$

Note that this implies $\left|A(\overline{S}, \neg q) - A(\hat{S}, \neg q)\right| \le \alpha$ as well. Then by taking a union bound over $\mathcal{Q}$, we know that $\left|A(\overline{S}, q) - A(\hat{S}, q)\right| \le \alpha$ holds for all $q \in \overline{\mathcal{Q}}$ with probability at least $1-\beta$. It follows that

$$\max_{q\in\overline{\mathcal{Q}}} A(\hat{S}, q) \le \max_{q\in\overline{\mathcal{Q}}} A(\overline{S}, q) + \alpha_0 \le \eta + \alpha_0,$$

which recovers the stated bound. □

The details of the algorithm are presented in Algorithm 4. To analyze the algorithm, we will start with establishing its privacy guarantee, which directly follows from the advanced

Algorithm 4 Oracle-Efficient Synthetic Data Release: OracleQuery

Given: Target privacy parameters $\varepsilon, \delta \in (0,1)$, a target failure probability β, a number of rounds T, accuracy parameters α_0, β_0, a weighted optimization oracle $\mathcal{O}$ for the class $\mathcal{Q}_{\text{dual}}$, a $(\varepsilon_0, \delta_0)$-differentially private (α_0, β_0)-accurate minimization oracle $\mathcal{O}_{\varepsilon_0,\delta_0}$ for class $\mathcal{Q}$ with parameters that satisfy

$$\varepsilon_0 = \frac{\varepsilon}{\sqrt{24T\ln(2/\delta)}}, \qquad \delta_0 \leq \frac{\delta}{4T}, \qquad \beta_0 = \frac{\beta}{4T}$$

Input: A dataset $S \in \mathcal{X}^n$.

1: Initialize $q^0 \in \overline{\mathcal{Q}}$ to be an arbitrary query
2: Let $N_{\alpha_0} = \frac{2\log(8|\mathcal{Q}|/\beta)}{\alpha_0^2}$
3: **for** $t = 1 \dots T$ **do**
4: Let S^t be a distribution defined by the sampling algorithm Context-FTPL $(\mathcal{Q}_{\text{dual}}, \mathcal{O}, \{q^0, \dots, q^{(t-1)}\})$ ▷ Data player's no-regret algorithm
5: Let $q^t = \text{PBR}(\varepsilon_0, \delta_0, \alpha_0, \beta_0, S, \mathcal{O}_{\varepsilon_0,\delta_0}, S^t)$ ▷ Query player's best response
6: **for** $j = 1, \dots, N_{\alpha_0}$: **do**
7: Draw τ from Unif($[T]$) and then draw x'_j from distribution S^τ

Output: the dataset $\hat{S} = \{x'_1, \dots, x'_{N_{\alpha_0}}\}$

composition of Dwork et al. (2010), the fact that each call to PBR by the Query player satisfies $(3\varepsilon_0, 2\delta_0)$-differential privacy (with ε_0 and δ_0 set according to Algorithm 4), and the fact that the rest of the algorithm can be viewed as a post-processing of these calls.

Lemma 4.4.11 (Privacy of OracleQuery)**.** OracleQuery *is oracle equivalent to an* (ε, δ)*-differentially private algorithm. If* $\mathcal{O}_{\varepsilon_0,\delta_0}$ *is robustly differentially private, then* OracleQuery *is* (ε, δ)*-robustly differentially private.*

Now to analyze the accuracy of OracleQuery, we will show that the average distribution $\overline{S}$ is part of an approximate minimax equilibrium, and so is its approximation $\hat{S}$.

Lemma 4.4.12 (Accuracy of OracleQuery)**.** *Suppose that* $\mathcal{O}_{\varepsilon_0,\delta_0}$ *is a weighted* $(\varepsilon_0, \delta_0)$*-differentially private* (α_0, β_0) *minimization oracle over the class* $\mathcal{Q}$*, where the parameters* ε_0*,* δ_0*, and* β_0 *are set according to Algorithm 4. Then* OracleQuery *is an* (α, β)*-accurate synthetic data generation algorithm for* $\mathcal{Q}$ *with*

$$\alpha \leq O\left(\alpha_0 + m_2^{3/4}\sqrt{\frac{\log(|\mathcal{X}|)}{T}} + \frac{\log(1/\beta_0)}{n\varepsilon_0}\right)$$

Proof. First, we will show that the average distribution $\overline{S}$ from the no-regret dynamics is an η-approximate minimax strategy, with

$$\eta \leq O\left(\alpha_0 + m_2^{3/4}\sqrt{\frac{\log(|\mathcal{X}|)}{T}} + \frac{\log(1/\beta_0)}{n\varepsilon_0}\right).$$

Recall that the average regret for the two players is bounded by

$$\mathrm{Reg}_D(T)/T \leq O(m_2^{3/4}\sqrt{\log|\mathcal{X}|/T}) \qquad \text{and,} \qquad \mathrm{Reg}_Q(T)/T \leq O\left(\alpha_0 + \frac{\log(1/\beta_0)}{n\varepsilon_0}\right),$$

with probability at least $1 - 3T\beta_0$. Then by Theorem 5.4.9, we know that $\overline{S}$ is an η-approximate minimax strategy, with probability at least $1 - 3\beta/4$. Let us condition on this event. Lastly, by the setting of N_{α_0} in Algorithm 4 and Lemma 4.4.10, we know that $\hat{S}$ is a $(\eta + \alpha_0)$, except with probability $\beta/4$. Then the stated bound follows directly from a union bound. □

Finally, we will consider three different instantiations of $\mathcal{O}_{\varepsilon_0,\delta_0}$. To optimize the error guarantee for each instantiation, we set the number of rounds T used in Algorithm 4 so that the regret of the Data player given by Theorem 4.4.8 is on the same order as the error of $\mathcal{O}_{\varepsilon_0,\delta_0}$. We first consider a differentially private oracle that matches the error guarantees of the generic private learner from Kasiviswanathan et al. (2011).

Corollary 4.4.13. *Suppose that $\mathcal{O}_{\varepsilon_0,0}$ is an $(\varepsilon_0, 0)$-differentially private (α_0, β_0)-accurate weighted minimization oracle, where α_0, β_0 are such that $\alpha_0 \leq O\left(\frac{\log(|\mathcal{Q}|/\beta_0)}{n\varepsilon_0}\right)$. Then* OracleQuery *with $T = \left\lceil \frac{n\varepsilon m_2^{3/4}\sqrt{\log(|\mathcal{X}|)}}{\log(|\mathcal{Q}|/\beta)\sqrt{\log(1/\delta)}} \right\rceil$ is an (α, β)-accurate synthetic data generation algorithm for $\mathcal{Q}$ with*

$$\alpha \leq \tilde{O}\left(\frac{m_2^{3/4}\sqrt{\log|\mathcal{X}|\log(1/\delta)}\log(|\mathcal{Q}|/\beta)}{n\varepsilon}\right)^{1/2}$$

where the $\tilde{O}$ hides logarithmic factors in m_2, n and $\log(|\mathcal{X}|)$.

Next, we will instantiate $\mathcal{O}_{\varepsilon_0,\delta_0}$ with the RSPM algorithm. Note that with this choice, although OracleQuery is oracle equivalent to a differentially private algorithm, it is not robustly differentially private.

Corollary 4.4.14. *When $\mathcal{O}_{\varepsilon_0,\delta_0}$ is instantiated with the Gaussian RSPM algorithm*, OracleQuery *with* $T = \left\lceil \frac{m_2^{3/4}\sqrt{\log(|\mathcal{X}|)}n\varepsilon}{m_1^{3/2}\sqrt{\log(m_1/\beta)}\ln(1/\delta)} \right\rceil$ *is an (α,β)-accurate synthetic data generation algorithm for $\mathcal{Q}$ with*

$$\alpha \leq \tilde{O}\left(\frac{m_1^{3/2}m_2^{3/4}\sqrt{\log(m_1/\beta)\log|\mathcal{X}|}\log(1/\delta)}{n\varepsilon}\right)^{1/2}$$

where the $\tilde{O}$ hides logarithmic factors in m_2, n and $\log(|\mathcal{X}|)$.

Finally, we instantiate the oracle $\mathcal{O}_{\varepsilon_0,\delta_0}$ with the **PRSMA** algorithm that uses the Laplace RSPM algorithm with a certifiable heuristic oracle.

Corollary 4.4.15. *When $\mathcal{O}_{\varepsilon_0,\delta_0}$ is instantiated with PRSMA algorithm (that internally uses Laplace RSPM as $\mathcal{A}_{\mathcal{O}^*}$), then for any $\beta > \delta$,* OracleQuery *with* $T = \left\lceil \left(\frac{m_2^{3/4}\sqrt{\log(|\mathcal{X}|)n\varepsilon}}{m_1^2+\sqrt{\log(|\mathcal{Q}|)}}\right)^{4/3} \right\rceil$ *is an (α,β)-accurate synthetic data generation algorithm for $\mathcal{Q}$ with*

$$\alpha \leq O\left(\frac{m_2^{1/4}\log^{1/6}|\mathcal{X}|\left(m_1^{4/3}+\log^{1/3}(|\mathcal{Q}|)\right)}{(n\varepsilon)^{1/3}}\right)\text{polylog}\left(m_1 1/\delta, 1/\varepsilon, \frac{1}{\beta-\delta}\right).$$

4.4.4. Example: Conjunctions

Here, we instantiate our bounds for a particular query class of interest: boolean conjunctions over the hypercube $\{0,1\}^d$. Constructing synthetic data for conjunctions (or equivalently producing a full *marginal table* for a dataset) has long been a challenge problem in differential privacy, subject to a long line of work Barak et al. (2007); Ullman and Vadhan (2010); Gupta et al. (2013); Kasiviswanathan et al. (2010); Thaler et al. (2012); Hardt et al. (2012); Feldman and Kothari (2014); Chandrasekaran et al. (2014), and is known to be computationally hard even for the special case of *two-way* conjunctions Ullman and Vadhan (2010). Our results in particular imply the first oracle efficient algorithm for generating synthetic data for all 2^d conjunctions. Other interesting classes satisfy all of the conditions

needed for our synthetic data generation algorithm to apply, including disjunctions, parities, and discrete halfspaces — see Appendix A.2.1 for details, and for how to allow negated variables in the class of conjunctions while preserving seperability.

Definition 4.4.16. *Given a subset of variables $S \subseteq [d]$, the boolean conjunction defined by S is the statistical query $q_S(x) = \wedge_{j \in S} x_j$. The set of boolean conjunctions $\mathcal{Q}_C$ defined over the hypercube $\mathcal{X} = \{0,1\}^d$ is:*

$$\mathcal{Q}_C = \{q_S \mid S \subseteq [d]\}$$

Boolean conjunctions are (d, d) dually-separable (see Appendix A.2.1 for the separator set).

Thus, we can instantiate Theorem 4.4.2 with (e.g.) the Gaussian RSPM algorithm, and obtain an oracle-efficient algorithm for generating synthetic data for all 2^d conjunctions that outputs a synthetic dataset S' that satisfies:

$$\max_{q \in \mathcal{Q}_C} |q(S) - q(S')| \leq \tilde{O}\left(\frac{d^{11/8}}{\sqrt{\varepsilon n}}\right)$$

4.5. A Barrier

In this paper, we give *oracle-efficient* private algorithms for learning and synthetic data generation for classes of queries $\mathcal{Q}$ that exhibit special structure: small universal identification sets. Because of information theoretic lower bounds for differentially private learning Bun et al. (2015); Alon et al. (2018), we know that these results cannot be extended to *all* learnable classes of queries $\mathcal{Q}$. But can they be extended to all classes of queries that are information theoretically learnable subject to differential privacy? Maybe — this is the most interesting question left open by our work. But here, we present a "barrier" illustrating a difficulty that one would have to overcome in trying to prove this result. Our argument has three parts:

1. First, we observe a folklore connection between differentially private learning and online learning: any differentially private empirical risk minimization algorithm $\mathcal{A}$

for a class $\mathcal{Q}$ that *always outputs the exact minimizer of a data-independent perturbation of the empirical risks* can also be used as a no-regret learning algorithm, using the "follow the perturbed leader" analysis of Kalai and Vempala Kalai and Vempala (2005). The per-round run-time of this algorithm is exactly equal to the run-time of $\mathcal{A}$.

2. Oracle-efficient no-regret learning algorithms are subject to a lower bound of Hazan and Koren Hazan and Koren (2016), that states that even given access to an oracle which solves optimization problems over a set of experts $\mathcal{Q}$ in unit time, there exist finite classes $\mathcal{Q}$ such that obtaining non-trivial regret guarantees requires total running time larger than poly($|Q|$). This implies a lower bound on the magnitude of the perturbations that an algorithm of the type described in (1) must use.

3. Finally, we observe for any finite class of hypotheses $\mathcal{Q}$, information theoretically, it is possible to solve the empirical risk minimization problem on a dataset of size T up to error $O(\frac{\log|\mathcal{Q}|}{\varepsilon T})$ using the generic learner from Kasiviswanathan et al. (2011). This implies a separation between the kinds of algorithms described in 1), and the (non-efficiently) achievable information theoretic bounds consistent with differential privacy.

We emphasize that oracle efficient algorithms for learning over $\mathcal{Q}$ have access to a non-private oracle which exactly solves the learning problem over $\mathcal{Q}$ — not an NP oracle, for which the situation is different (see the discussion in Section 4.6).

First we define the class of mechanisms our barrier result applies to:

Definition 4.5.1. *We say that an (ε, δ)-differentially private learning algorithm $\mathcal{A} : \mathcal{X}^n \to \mathcal{Q}$ for $\mathcal{Q}$ is a perturbed Empirical Risk Minimizer (pERM) if there is some distribution $\mathcal{D}_{\varepsilon,\delta}$ (defined independently of the data S) over perturbations $Z \in \mathbb{R}^{|\mathcal{Q}|}$ such that on input $S \in \mathcal{X}^n$, $\mathcal{A}$ outputs:*

$$\mathcal{A}(S) = \arg\min_{q \in \mathcal{Q}} \left(n \cdot q(S) + Z_q \right)$$

where $Z \sim \mathcal{D}_{\varepsilon,\delta}$.

We note that many algorithms are pERM algorithms. The most obvious example is *report-noisy-min*, in which each $Z_q \sim \text{Lap}(1/\varepsilon)$ independently. The exponential mechanism instantiated with empirical loss as its quality score (i.e. the generic learner of Kasiviswanathan et al. (2011)) is also a pERM algorithm, in which each Z_q is drawn independently from a Gumbel distribution Dwork and Roth (2014b). But note that the coordinates Z_q need not be drawn independently: The oracle-efficient RSPM algorithm we give in Section 4.3 is also a pERM algorithm, in which the perturbations Z_q are introduced in an implicit (correlated) way by perturbing the dataset itself. And it is natural to imagine that many algorithms that employ weighted optimization oracles — which after all solve an exact minimization problem — will fall into this class. The expected error guarantees of these algorithms are proven by bounding $\mathbb{E}[\|Z\|_\infty]$, which is typically a tight bound.

We now briefly recall the online learning setting. Let $\mathcal{Q}$ be an arbitrary class of functions $q : \mathcal{X} \to [0,1]$. In rounds $t = 1, \ldots, T$, the learner selects a function $q^t \in \mathcal{Q}$, and an (adaptive) adversary selects an example $x^t \in \mathcal{X}$, as a function of the sequence $(q^1, x^1, \ldots, q^{t-1}, x^{t-1})$. The learner incurs a loss of $\ell^t = q^t(x^t)$. A standard objective is to minimize the *expected average regret*:

$$R(T) = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} q^t(x^t)\right] - \min_{q\in\mathcal{Q}} \frac{1}{T}\sum_{t=1}^{T} q(x^t)$$

where the expectation is taken over the randomness of the learner. A weighted optimization oracle in the online learning setting is exactly the same thing as it is in our setting: Given a weighted dataset (S, w), it returns $\arg\min_{q\in\mathcal{Q}} \sum_{x_i \in S} w_i \cdot q(x_i)$.

A natural way to try to use a private learning algorithm in the online learning setting is just to run it at each round t on the dataset defined on the set of data points observed so far: $S_t = \{x^1, \ldots, x^{t-1}\}$.

Definition 4.5.2. *Follow the Private Leader, instantiated with $\mathcal{A}$, is the online learning algorithm that at every round t selects $q^t = \mathcal{A}(S_t)$.*

The follow the private leader algorithm instantiated with $\mathcal{A}$ has a controllable regret bound whenever $\mathcal{A}$ is a differentially private pERM algorithm. The following theorem is folklore, but follows essentially from the original analysis of "follow the perturbed leader" by Kalai and Vempala Kalai and Vempala (2005). See e.g. the lecture notes from Roth and Smith (2017) or Abernethy et al. (2017) for an example of this analysis cast in the language of differential privacy. We include a proof in Appendix A.2.4 for completeness.

Theorem 4.5.3. *Let $\varepsilon, \delta \in (0,1)$ and let $\mathcal{A}$ be an (ε, δ) differentially private pERM algorithm for query class $\mathcal{Q}$, with perturbation distribution $\mathcal{D}_{(\varepsilon,\delta)}$. Then Follow the Private Leader instantiated with $\mathcal{A}$ has expected regret bounded by:*

$$R(T) \leq O\left(\varepsilon + \delta + \frac{\mathbb{E}_{Z\sim\mathcal{D}_{\varepsilon,\delta}}[\|Z\|_\infty]}{T}\right)$$

Note that the regret is controlled by $\mathbb{E}_{Z\sim\mathcal{D}_{\varepsilon,\delta}}[\|Z\|_\infty]$, which also controls the error of $\mathcal{A}$ as a learning algorithm.

We wish to exploit a lower bound on the running time of oracle efficient online learners over arbitrary sets $\mathcal{Q}$ due to Hazan and Koren Hazan and Koren (2016):

Theorem 4.5.4 (Hazan and Koren (2016))**.** *For every algorithm with access to a weighted optimization oracle $\mathcal{O}$, there exists a class of functions $\mathcal{Q}$ such that the algorithm cannot guarantee that its expected average regret will be smaller than* 1/16 *in total time less than $O\left(\sqrt{|\mathcal{Q}|}/\log^3(|\mathcal{Q}|)\right)$.*

Here, it is assumed that calls to the oracle $\mathcal{O}$ can be carried out in unit time, and *total* time refers to the cumulative time over all T rounds of interaction. Hence, if $\mathcal{A}$ is oracle-efficient — i.e. it runs in time $f(t) = \text{poly}(t, \log|\mathcal{Q}|)$ when given as input a dataset of size t, the *total* run time of follow the private leader instantiated with $\mathcal{A}$ is: $\sum_{t=1}^{T} f(t) \leq T \cdot f(T) = \text{poly}(T, \log|\mathcal{Q}|)$.

This theorem is almost what we want — except that the order of quantifiers is reversed. It in principle leaves open the possibility that for every class $\mathcal{Q}$, there is a different oracle efficient algorithm (tailored to the class) that can efficiently obtain low regret. After all, our

RSPM algorithm is non-uniform in this way — for each new class of functions $\mathcal{Q}$, it must be instantiated with a separator set for that class.

Via a min-max argument together with an equilibrium sparsification technique, we can give a version of the lower bound of Hazan and Koren (2016) that has the order of quantifiers we want — see Appendix A.2.4 for the proof.

Theorem 4.5.5. *For any d, there is a fixed finite class of statistical queries $\mathcal{Q}$ of size $|\mathcal{Q}| = N = 2^d$ defined over a data universe of size $|\mathcal{X}| = O(N^5 \log^2 N)$ such that for every online learning algorithm with access to a weighted optimization oracle for $\mathcal{Q}$, it cannot guarantee that its expected average regret will be $o(1)$ in total time less than $\Omega(\sqrt{N}/\log^3(N))$.*

Theorem 4.5.5 therefore implies that follow the private leader, when instantiated with any oracle-efficient differentially private pERM algorithm $\mathcal{A}$ cannot obtain diminishing regret $R(T) = o(1)$ unless the number of rounds $T = \Omega(|Q|^c)$ for some $c > 0$. In combination with Theorem 4.5.3, this implies our barrier result:

Theorem 4.5.6. *Any oracle efficient (i.e. running in time poly$(n, \log|\mathcal{Q}|)$) (ε, δ)-differentially private pERM algorithm instantiated with a weighted optimization oracle for the query class $\mathcal{Q}$ defined in Theorem 4.5.5, with perturbation distribution $\mathcal{D}_{(\varepsilon,\delta)}$ must be such that for every $(\varepsilon + \delta) = o(1)$:*

$$\mathbb{E}_{Z \sim \mathcal{D}_{(\varepsilon,\delta)}}[\|Z\|_\infty] \geq \Omega(|\mathcal{Q}|^c)$$

for some constant $c > 0$.

If the accuracy guarantee of $\mathcal{A}$ is proportional to $\mathbb{E}_{Z \sim \mathcal{D}_{(\varepsilon,\delta)}}[\|Z\|_\infty]$ (as it is for all pERM algorithms that we know of), this means that there exist finite classes of statistical queries $\mathcal{Q}$ such that no oracle-efficient algorithm can obtain non-trivial error unless the dataset size $n \geq \text{poly}(|\mathcal{Q}|)$. Of course, if $n \geq \text{poly}(|\mathcal{Q}|)$, then algorithms such as report-noisy-min and the exponential mechanism can be run in polynomial time.

This is in contrast with what we can obtain via the generic (inefficient) private learner of Kasiviswanathan et al. (2011), which obtains expected error $O\left(\frac{\log|Q|}{\varepsilon n}\right)$, which is non-trivial

whenever $n = \Omega\left(\frac{\log|\mathcal{Q}|}{\varepsilon}\right)$. Similarly, because we show in Theorem 4.5.5 that the hard class $\mathcal{Q}$ can be taken to have universe size $\mathcal{X} = \text{poly}(\mathcal{Q})$, this means that information theoretically, it is even possible to privately solve the (harder) problem of α-accurate synthetic data for $\mathcal{Q}$ for $\alpha = O\left(\left(\frac{\log^2|\mathcal{Q}|}{\varepsilon n}\right)^{1/3}\right)$ using the (inefficient) synthetic data generation algorithm of Blum et al. (2013). This is non-trivial whenever $n = \Omega\left(\frac{\log^2|\mathcal{Q}|}{\varepsilon}\right)$. In contrast, our barrier result states is that *if* there exists an oracle-efficient learner $\mathcal{A}$ for this class $\mathcal{Q}$ that has polynomially related sample complexity to what is obtainable absent a guarantee of oracle efficiency, then $\mathcal{A}$ must either:

1. Not be a pERM algorithm, or:
2. Have expected error that is $O\left(\frac{\text{poly}(\log \mathbb{E}_{Z\sim\mathcal{D}_{(\varepsilon,\delta)}}[\|Z\|_\infty])}{n}\right)$.

Condition 2. seems especially implausible, as for every pERM we are aware of, $\mathbb{E}_{Z\sim\mathcal{D}_{(\varepsilon,\delta)}}[\|Z\|_\infty]$ is a tight bound (up to log factors) on its expected error. In particular, this barrier implies that there is no oracle efficient algorithm for *sampling* from the exponential mechanism distribution used in the generic learner of Kasiviswanathan et al. (2011) for arbitrary query classes $\mathcal{Q}$.

4.6. Conclusion and Open Questions

In this paper, we have initiated the systematic study of the power of *oracle-efficient* differentially private algorithms, and have made the distinction between oracle-dependent non-robust differential privacy and robust differential privacy. This is a new direction that suggests a number of fascinating open questions. In our opinion, the most interesting of these is:

> "Can every learning and synthetic data generation problem that is solvable subject to differential privacy be solved with an oracle-efficient (robustly) differentially private algorithm, with only a polynomial blow-up in sample complexity?"

It remains an open question whether or not finite Littlestone dimension characterizes private learnability (it is known that infinite Littlestone dimension precludes private learnability Alon et al. (2018)) — and so one avenue towards resolving both open questions in the affirmative simultaneously would be to show that finite Littlestone dimension can be leveraged to obtain oracle-efficient differentially private learning algorithms.

However, because of our barrier result, we conjecture that the set of query classes that are privately learnable in an oracle-efficient manner is a *strict subset* of the set that are privately learnable. If this is so, can we precisely characterize this set? What is the right structural property, and is it more general than the sufficient condition of having small universal identification sets that we have discovered?

Even restricting attention to query classes with universal identification sets of size m, there are interesting quantitative questions. The Gaussian version of our RSPM algorithm efficiently obtains error that scales as $m^{3/2}$, but information-theoretically, it is possible to obtain error scaling only linearly with m. Is this optimal error rate possible to obtain in an oracle-efficient manner, or is the $\sqrt{m}$ error overhead that comes with our approach necessary for oracle efficiency?

Our **PRSMA** algorithm shows how to generically reduce from an oracle-dependent guarantee of differential privacy to a guarantee of robust differential privacy — *but at a cost*, both in terms of running time, and in terms of error. Are these costs necessary? Without further assumptions on the construction of the oracle, it seems difficult to avoid the $O(1/\delta)$-overhead in running time, but perhaps there are natural assumptions that can be placed on the failure-mode of the oracle that can avoid this. It is less clear whether the error overhead that we introduce — by running the original algorithm on an ε fraction of the dataset, with a privacy parameter $\varepsilon' \approx 1/\sqrt{\varepsilon n}$ — is necessary. Doing this is a key feature of our algorithm and analysis, because we take advantage of the fact that differentially private algorithms are actually *distributionally private* when ε' is set this small — but perhaps it can be avoided entirely with a different approach.

Our barrier result takes advantage of a connection between differentially private learnability and online learnability. Because private pERM algorithms can be used efficiently as no-regret learning algorithms, they are subject to the lower bounds on oracle-efficient online learning proven in Hazan and Koren (2016). But perhaps the connection between differentially private learnability and online learnability runs deeper. Can *every* differentially private learning algorithm be used in a black box manner to efficiently obtain a no-regret learning algorithm? Note that it is already known that private learnability implies finite Littlestone dimension, so the open question here concerns whether there is an *efficient blackbox* reduction from private ERM algorithms to online learning algorithms. If true, this would convert our barrier for pERM algorithms into a full lower-bound for oracle-efficient private learning algorithms generally.

Finally, a more open ended question — that applies both to our work and to work on oracle efficiency in machine learning more generally — concerns how to refine the model of oracle efficiency. Ideally, the learning problems fed to the oracle should be "natural" — e.g. a small perturbation or re-weighting of the original (non-private) learning problem, as is the case for the algorithms we present in our paper. This is desirable because presumably we believe that the heuristics which can solve hard learning problems in practice work for "natural" instances, rather than arbitrary problems. However, the definition for oracle efficiency that we use in this paper allows for un-natural algorithms. For example, it is possible to show that the problem of sampling from the exponential mechanism of McSherry and Talwar (2007) defined by rational valued quality scores that are efficiently computable lies in $\mathbf{BPP}^{\mathbf{NP}}$ — in other words, the sampling can be done in polynomial time given access to an oracle for solving circuit-satisfiability problems[4]. This implies in particular, that there exists an oracle efficient algorithm (as we have defined them) for any NP hard

[4]This construction is due to Jonathan Ullman and Salil Vadhan (personal communication). It starts from the ability to sample uniformly at random amongst the set of satisfying assignments of an arbitrary polynomially sized boolean circuit given an NP oracle, using the algorithm of Bellare et al. (2000). For any distribution $\mathcal{P}$ such that there is a polynomially sized circuit C for which the relative probability mass on any discrete input x can be computed by $C(x)$, we can construct a boolean circuit C' that computes for bounded bit-length rational numbers w: $C'(x, w) = 1$ if $C(x) \geq w$. The marginal distribution on elements x when sampling uniformly at random from the satisfying assignments of this circuit is $\mathcal{P}$.

learning problem — because the learning oracle can be used as an arbitrary **NP** oracle via gadget reductions[5]. The same logic implies that there are oracle efficient no-regret learning algorithms for any class of experts for which offline optimization is NP hard — because an NP oracle can be used to sample from the multiplicative weights distribution. But these kinds of gadget reductions seem to be an abuse of the model of oracle efficiency, which currently reduces to all of $\mathbf{BPP}^{\mathbf{NP}}$ when the given oracle is solving an NP hard problem[6]. Ambitiously, might there be a refinement of the model of oracle efficiency that requires one to prove a utility theorem along the following lines: assuming an oracle which can with high probability solve learning problems drawn from the actual data distribution, the oracle efficient algorithm will (with slightly lower probability) solve the private learning problem when the underlying instance is drawn from the same distribution. Theorems of this sort would be of great interest, and would (presumably) rule out "unnatural" algorithms relying on gadget reductions.

[5]Note that this procedure is not *robustly* differentially private, since sampling from the correct distribution occurs only if the oracle does not fail. But it could be fed into our **PRSMA** algorithm to obtain robust privacy. It also does not solve synthetic data generation oracle efficiently because the quality score used for synthetic data generation in Blum et al. (2013) is not computable by a polynomially sized circuit generally.

[6]This does not contradict the lower bound of Hazan and Koren (2016) for oracle efficient online learning, or our barrier result/conjectured separation in the case of private learning algorithms. This is because oracles solving problems that don't have polynomial time algorithms, but *are not NP hard* cannot be used to encode the arbitrary circuit-SAT instances needed to implement an NP oracle.

CHAPTER 5

Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness

5.1. Introduction

The semantics of the statistical notions of fairness discussed in Chapter 1 would be significantly stronger if they were defined over a large number of *subgroups*, thus permitting a rich middle ground between fairness only for a small number of coarse pre-defined groups, and the strong assumptions needed for fairness at the individual level. Consider the kind of *fairness gerrymandering* that can occur when we only look for unfairness over a small number of pre-defined groups:

Example 5.1.1. *Imagine a setting with two binary features, corresponding to race (say black and white) and gender (say male and female), both of which are distributed independently and uniformly at random in a population. Consider a classifier that labels an example positive if and only if it corresponds to a black man, or a white woman. Then the classifier will appear to be equitable when one considers either protected attribute alone, in the sense that it labels both men and women as positive 50% of the time, and labels both black and white individuals as positive 50% of the time. But if one looks at any* conjunction *of the two attributes (such as black women), then it is apparent that the classifier maximally violates the statistical parity fairness constraint. Similarly, if examples have a binary label that is also distributed uniformly at random, and independently from the features, the classifier will satisfy equal opportunity fairness with respect to either protected attribute alone, even though it maximally violates it with respect to conjunctions of two attributes.*

We remark that the issue raised by this toy example is not merely hypothetical. In our experiments in Section 5.5, we show that similar violations of fairness on subgroups of the pre-defined groups can result from the application of standard machine learning methods applied to real datasets. To avoid such problems, we would like to be able to satisfy a

fairness constraint not just for the small number of protected groups defined by single protected attributes, but for a combinatorially large or even infinite collection of structured subgroups definable over protected attributes.

In this chapter, we consider the problem of *auditing* binary classifiers for equal opportunity and statistical parity, and the problem of *learning* classifiers subject to these constraints, when the number of protected groups is large. There are exponentially many ways of carving up a population into subgroups, and we cannot necessarily identify a small number of these *a priori* as the only ones we need to be concerned about. At the same time, we cannot insist on any notion of statistical fairness for *every* subgroup of the population: for example, any imperfect classifier could be accused of being unfair to the subgroup of individuals defined ex-post as the set of individuals it misclassified. This simply corresponds to "overfitting" a fairness constraint. We note that the individual fairness definition of Joseph et al. (2016) (when restricted to the binary classification setting) can be viewed as asking for equalized false positive rates across the singleton subgroups, containing just one individual each[1] — but naturally, in order to achieve this strong definition of fairness, Joseph et al. (2016) have to make structural assumptions about the form of the ground truth. It is, however, sensible to ask for fairness for large *structured* subsets of individuals: so long as these subsets have a bounded VC dimension, the *statistical* problem of learning and auditing fair classifiers is easy, so long as the dataset is sufficiently large. This can be viewed as an interpolation between equal opportunity fairness and the individual "weakly meritocratic" fairness definition from Joseph et al. (2016), that does not require making any assumptions about the ground truth. Our investigation focuses on the computational challenges, both in theory and in practice.

5.1.1. Our Results

Briefly, our contributions are:

[1] It also asks for equalized false negative rates, and that the false positive rate is smaller than the true positive rate. Here, the randomness in the "rates" is taken entirely over the randomness of the classifier.

- Formalization of the problem of auditing and learning classifiers for fairness with respect to rich classes of subgroups $\mathcal{G}$.

- Results proving (under certain assumptions) the computational equivalence of auditing $\mathcal{G}$ and (weak) agnostic learning of $\mathcal{G}$. While these results imply theoretical intractability of auditing for some natural classes $\mathcal{G}$, they also suggest that practical machine learning heuristics can be applied to the auditing problem.

- Provably convergent algorithms for learning classifiers that are fair with respect to $\mathcal{G}$, based on a formulation as a two-player zero-sum game between a Learner (the primal player) and an Auditor (the dual player). We provide two different algorithms, both of which are based on solving for the equilibrium of this game. The first provably converges in a polynomial number of steps and is based on simulation of the game dynamics when the Learner uses *Follow the Perturbed Leader* and the Auditor uses best response; the second is only guaranteed to converge asymptotically but is computationally simpler, and involves both players using *Fictitious Play*.

- An implementation and empirical evaluation of the Fictitious Play algorithm demonstrating its effectiveness on a real dataset in which subgroup fairness is a concern.

In more detail, we start by studying the computational challenge of simply *checking* whether a given classifier satisfies equal opportunity and statistical parity. Doing this in time linear in the number of protected groups is simple: for each protected group, we need only estimate a single expectation. However, when there are many different protected attributes which can be combined to define the protected groups, their number is combinatorially large[2].

[2]For example, as discussed in a recent Propublica investigation (Angwin and Grassegger, 2017), Facebook policy protects groups against hate speech if the group is definable as a *conjunction* of protected attributes. Under the Facebook schema, "race" and "gender" are both protected attributes, and so the Facebook policy protects "black women" as a distinct class, separately from black people and women. When there are d protected attributes, there are 2^d protected groups. As a statistical estimation problem, this is not a large obstacle — we can estimate 2^d expectations to error ε so long as our data set has size $O(d/\varepsilon^2)$, but there is now a computational problem.

We model the problem by specifying a class of functions $\mathcal{G}$ defined over a set of d protected attributes. $\mathcal{G}$ defines a set of protected subgroups. Each function $g \in \mathcal{G}$ corresponds to the protected subgroup $\{x : g_i(x) = 1\}$[3]. The first result of this chapter is that for both equal opportunity and statistical parity, the computational problem of *checking* whether a classifier or decision-making algorithm D violates statistical fairness with respect to the set of protected groups $\mathcal{G}$ is equivalent to the problem of *agnostically learning* $\mathcal{G}$ (Kearns et al., 1994), in a strong and distribution-specific sense. This equivalence has two implications:

1. First, it allows us to import *computational hardness* results from the learning theory literature. Agnostic learning turns out to be computationally hard in the worst case, even for extremely simple classes of functions $\mathcal{G}$ (like boolean conjunctions and linear threshold functions). As a result, we can conclude that auditing a classifier D for statistical fairness violations with respect to a class $\mathcal{G}$ is also computationally hard. This means we should not expect to find a polynomial time algorithm that is always guaranteed to solve the auditing problem.

2. However, in practice, various learning heuristics (like boosting, logistic regression, SVMs, backpropagation for neural networks, etc.) are commonly used to learn accurate classifiers which are known to be hard to learn in the worst case. The equivalence we show between agnostic learning and auditing is *distribution specific* — that is, if on a particular data set, a heuristic learning algorithm can solve the agnostic learning problem (on an appropriately defined subset of the data), it can be used also to solve the auditing problem on the same data set.

These results appear in Section 5.3.

[3]For example, in the case of Facebook's policy, the protected attributes include "race, sex, gender identity, religious affiliation, national origin, ethnicity, sexual orientation and serious disability/disease" (Angwin and Grassegger, 2017), and $\mathcal{G}$ represents the class of boolean conjunctions. In other words, a group defined by individuals having any *subset* of values for the protected attributes is protected.

Next, we consider the problem of *learning* a classifier that equalizes false positive or negative rates across all (possibly infinitely many) sub-groups, defined by a class of functions $\mathcal{G}$. As per the reductions described above, this problem is computationally hard in the worst case.

However, under the assumption that we have an efficient oracles which solves the *agnostic learning* problem, we give and analyze algorithms for this problem based on a game-theoretic formulation. We first prove that the optimal fair classifier can be found as the equilibrium of a two-player, zero-sum game, in which the (pure) strategy space of the "Learner" player corresponds to classifiers in $\mathcal{H}$, and the (pure) strategy space of the "Auditor" player corresponds to subgroups defined by $\mathcal{G}$. The best response problems for the two players correspond to agnostic learning and auditing, respectively. We show that both problems can be solved with a single call to a *cost sensitive classification oracle*, which is equivalent to an agnostic learning oracle. We then draw on extant theory for learning in games and no-regret algorithms to derive two different algorithms based on simulating game play in this formulation. In the first, the Learner employs the well-studied *Follow the Perturbed Leader (FTPL)* algorithm on an appropriate linearization of its best-response problem, while the Auditor approximately best-responds to the distribution over classifiers of the Learner at each step. Since FTPL has a no-regret guarantee, we obtain an algorithm that provably converges in a polynomial number of steps.

While it enjoys strong provable guarantees, this first algorithm is randomized (due to the noise added by FTPL), and the best-response step for the Auditor is polynomial time but computationally expensive. We thus propose a second algorithm that is deterministic, simpler and faster per step, based on both players adopting the Fictitious Play learning dynamic. This algorithm has weaker theoretical guarantees: it has provable convergence only asymptotically, and not in a polynomial number of steps — but is more practical and converges rapidly in practice. The derivation of these algorithms (and their guarantees) appear in Section 5.4.

Finally, we implement the Fictitious Play algorithm and demonstrate its practicality by efficiently learning classifiers that approximately equalize false positive rates across any group definable by a linear threshold function on 18 protected attributes in the "Communities and Crime" dataset. We use simple, fast regression algorithms as heuristics to implement agnostic learning oracles, and (via our reduction from agnostic learning to auditing) auditing oracles. Our results suggest that it is possible in practice to learn fair classifiers with respect to a large class of subgroups that still achieve non-trivial error. Full details are contained in Section 5.5, and for a substantially more comprehensive empirical investigation of our method we direct the interested reader to Kearns et al. (2018).

5.1.2. Further Related Work

Independent of our work, Hébert-Johnson et al. (2017) also consider a related and complementary notion of fairness that they call "multicalibration". In settings in which one wishes to train a real-valued predictor, multicalibration can be considered the "calibration" analogue for the definitions of subgroup fairness that we give for false positive rates, false negative rates, and classification rates. For a real-valued predictor, calibration informally requires that for every value $v \in [0,1]$ predicted by an algorithm, the fraction of individuals who truly have a positive label in the subset of individuals on which the algorithm predicted v should be approximately equal to v. Multicalibration asks for approximate calibration on every set defined implicitly by some circuit in a set $\mathcal{G}$. Hébert-Johnson et al. (2017) give an algorithmic result that is analogous to the one we give for learning subgroup fair classifiers: a polynomial time algorithm for learning a multi-calibrated predictor, given an agnostic learning algorithm for $\mathcal{G}$. In addition to giving a polynomial-time algorithm, we also give a practical variant of our algorithm (which is however only guaranteed to converge in the limit) that we use to conduct empirical experiments on real data.

Thematically, the most closely related piece of prior work is Zhang and Neill (2016), who also aim to audit classification algorithms for discrimination in subgroups that have not been pre-defined. Our work differs from theirs in a number of important ways. First, we

audit the algorithm for common measures of statistical unfairness, whereas Zhang and Neill (2016) design a new measure compatible with their particular algorithmic technique. Second, we give a formal analysis of our algorithm. Finally, we audit with respect to subgroups defined by a class of functions $\mathcal{G}$, which we can take to have bounded VC dimension, which allows us to give formal out-of-sample guarantees. Zhang and Neill (2016) attempt to audit with respect to *all possible* sub-groups, which introduces a severe multiple-hypothesis testing problem, and risks overfitting. Most importantly we give actionable algorithms for learning subgroup fair classifiers, whereas Zhang and Neill (2016) restrict attention to auditing.

Technically, the most closely related piece of work (and from which we take inspiration for our algorithm in Section 5.4) is Agarwal et al. (2017), who show that given access to an agnostic learning oracle for a class $\mathcal{H}$, there is an efficient algorithm to find the lowest-error distribution over classifiers in $\mathcal{H}$ subject to equalizing false positive rates across polynomially many subgroups. Their algorithm can be viewed as solving the same zero-sum game that we solve, but in which the "subgroup" player plays gradient descent over his pure strategies, one for each sub-group. This ceases to be an efficient or practical algorithm when the number of subgroups is large, as is our case. Our main insight is that an agnostic learning oracle is sufficient to have the both players play "fictitious play", and that there is a transformation of the best response problem such that an agnostic learning algorithm is enough to efficiently implement follow the perturbed leader.

There is also other work showing computational hardness for fair learning problems. Most notably, Woodworth et al. (2017) show that finding a linear threshold classifier that approximately minimizes hinge loss subject to equalizing false positive rates across populations is computationally hard (assuming that refuting a random k-XOR formula is hard). In contrast, we show that even *checking* whether a classifier satisfies a false positive rate constraint on a particular data set is computationally hard (if the number of subgroups on which fairness is desired is too large to enumerate).

5.2. Model and Preliminaries

We model each individual as being described by a tuple $((x, x'), y)$, where $x \in \mathcal{X}$ denotes a vector of *protected attributes*, $x' \in \mathcal{X}'$ denotes a vector of *unprotected attributes*, and $y \in \{0, 1\}$ denotes a label. Note that in our formulation, an auditing algorithm not only may not see the unprotected attributes x', it may not even be aware of their existence. For example, x' may represent proprietary features or consumer data purchased by a credit scoring company.

We will write $X = (x, x')$ to denote the joint feature vector. We assume that points (X, y) are drawn i.i.d. from an unknown distribution $\mathcal{P}$. Let D be a decision making algorithm, and let $D(X)$ denote the (possibly randomized) decision induced by D on individual (X, y). We restrict attention in this paper to the case in which D makes a binary classification decision: $D(X) \in \{0, 1\}$. Thus we alternately refer to D as a classifier. When *auditing* a fixed classifier D, it will be helpful to make reference to the distribution over examples (X, y) together with their induced classification $D(X)$. Let $P_{\text{audit}}(D)$ denote the induced *target joint distribution* over the tuple $(x, y, D(X))$ that results from sampling $(x, x', y) \sim \mathcal{P}$, and providing x, the true label y, and the classification $D(X) = D(x, x')$ but not the unprotected attributes x'. Note that the randomness here is over both the randomness of $\mathcal{P}$, and the potential randomness of the classifier D.

We will be concerned with learning and auditing classifiers D satisfying two common statistical fairness constraints: equality of classification rates (also known as statistical parity), and equality of false positive rates (also known as equal opportunity). Auditing for equality of false negative rates is symmetric and so we do not explicitly consider it. Each fairness constraint is defined with respect to a set of protected groups. We define sets of protected groups via a family of indicator functions $\mathcal{F}$ for those groups, defined over protected attributes. Each $g : \mathcal{X} \to \{0, 1\} \in \mathcal{F}$ has the semantics that $g(x) = 1$ indicates that an individual with protected features x is in group g.

Definition 5.2.1 (Statistical Parity (SP) Subgroup Fairness)**.** *Fix any classifier D, distribution $\mathcal{P}$, collection of group indicators $\mathcal{G}$, and parameter $\gamma \in [0,1]$. For each $g \in \mathcal{G}$, define*

$$\alpha_{SP}(g,\mathcal{P}) = \Pr_{\mathcal{P}}[g(x)=1] \quad \textit{and,} \quad \beta_{SP}(g,D,\mathcal{P}) = |\mathrm{SP}(D) - \mathrm{SP}(D,g)|,$$

where $\mathrm{SP}(D) = \Pr_{\mathcal{P},D}[D(X)=1]$ *and* $\mathrm{SP}(D,g) = \Pr_{\mathcal{P},D}[D(X)=1|g(x)=1]$ *denote the overall acceptance rate of D and the acceptance rate of D on group g respectively. We say that D satisfies* γ-statistical parity (SP) Fairness *with respect to $\mathcal{P}$ and $\mathcal{F}$ if for every $g \in \mathcal{F}$*

$$\alpha_{SP}(g,\mathcal{P})\ \beta_{SP}(g,D,\mathcal{P}) \leq \gamma.$$

We will sometimes refer to $\mathrm{SP}(D)$ *as the* SP base rate.

Remark 5.2.2. *Note that our definition references two approximation parameters, both of which are important. We are allowed to ignore a group g if it (or its complement) represent only a small fraction of the total probability mass. The parameter α governs how small a fraction of the population we are allowed to ignore. Similarly, we do not require that the probability of a positive classification in every subgroup is exactly equal to the base rate, but instead allow deviations up to β. Both of these approximation parameters are necessary from a statistical estimation perspective. We control both of them with a single parameter γ.*

Definition 5.2.3 (False Positive (FP) Subgroup Fairness)**.** *Fix any classifier D, distribution $\mathcal{P}$, collection of group indicators $\mathcal{F}$, and parameter $\gamma \in [0,1]$. For each $g \in \mathcal{G}$, define*

$$\alpha_{FP}(g,\mathcal{P}) = \Pr_{\mathcal{P}}[g(x)=1, y=0] \quad \textit{and,} \quad \beta_{FP}(g,D,\mathcal{P}) = |\mathrm{FP}(D) - \mathrm{FP}(D,g)|$$

where $\mathrm{FP}(D) = \Pr_{D,\mathcal{P}}[D(X)=1 \mid y=0]$ *and* $\mathrm{FP}(D,g) = \Pr_{D,\mathcal{P}}[D(X)=1 \mid g(x)=1, y=0]$ *denote the overall false-positive rate of D and the false-positive rate of D on group g respectively.*

We say D satisfies γ-False Positive (FP) Fairness *with respect to $\mathcal{P}$ and $\mathcal{F}$ if for every $g \in \mathcal{G}$*

$$\alpha_{FP}(g,\mathcal{P})\ \beta_{FP}(g,D,\mathcal{P}) \leq \gamma.$$

We will sometimes refer to $\mathrm{FP}(D)$ *FP-base rate.*

Remark 5.2.4. *This definition is symmetric to the definition of statistical parity fairness, except that the parameter* α *is now used to exclude any group* g *such that* negative examples *(*$y = 0$*) from* g *(or its complement) have probability mass less than* α*. This is again necessary from a statistical estimation perspective.*

For either statistical parity and false positive fairness, if the algorithm D fails to satisfy the γ-fairness condition, then we say that D is γ-*unfair* with respect to $\mathcal{P}$ and $\mathcal{F}$. We call any subgroup g which witnesses this unfairness an γ-*unfair certificate* for $(D, \mathcal{P})$.

An *auditing algorithm* for a notion of fairness is given sample access to $P_{\text{audit}}(D)$ for some classifier D. It will either deem D to be fair with respect to $\mathcal{P}$, or will else produce a certificate of unfairness.

Definition 5.2.5 (Auditing Algorithm)**.** *Fix a notion of fairness (either statistical parity or false positive fairness), a collection of group indicators* $\mathcal{F}$ *over the protected features, and any* $\delta, \gamma, \gamma' \in (0,1)$ *such that* $\gamma' \leq \gamma$*. A* (γ, γ')*-auditing algorithm for* $\mathcal{F}$ *with respect to distribution* $\mathcal{P}$ *is an algorithm* $\mathcal{A}$ *such that for any classifier* D*, when given access the distribution* $P_{\text{audit}}(D)$*,* $\mathcal{A}$ *runs in time* $\mathrm{poly}(1/\gamma', \log(1/\delta))$*, and with probability* $1 - \delta$*, outputs a* γ'*-unfair certificate for* D *whenever* D *is* γ*-unfair with respect to* $\mathcal{P}$ *and* $\mathcal{F}$*. If* D *is* γ'*-fair,* $\mathcal{A}$ *will output "fair".*

As we will show, our definition of auditing is closely related to weak agnostic learning. Throughout the paper we will invoke the definition of cost-sensitive classification oracles and lemma that follows in Subsection 2.2.

Follow the Perturbed Leader. We will make use of the *Follow the Perturbed Leader (FTPL)* algorithm as a no-regret learner for online linear optimization problems (Kalai and Vempala, 2005). To formalize the algorithm, consider $\mathcal{S} \subset \{0,1\}^d$ to be a set of "actions" for a learner in an online decision problem. The learner interacts with an adversary over T rounds, and in each round t, the learner (randomly) chooses some action $a^t \in \mathcal{S}$, and the adversary chooses a loss vector $\ell^t \in [-M, M]^d$. The learner incurs a loss of $\langle \ell^t, a^t \rangle$ at round t.

FTPL is a simple algorithm that in each round perturbs the cumulative loss vector over the previous rounds $\overline{\ell} = \sum_{s<t} \ell^s$, and chooses the action that minimizes loss with respect to the perturbed cumulative loss vector. We present the full algorithm in Algorithm 5, and its formal guarantee in Theorem 5.2.6.

Algorithm 5 Follow the Perturbed Leader (FTPL) Algorithm

Input: Loss bound M, action set $\mathcal{S} \in \{0,1\}^d$
Initialize: Let $\eta = (1/M)\sqrt{\frac{1}{\sqrt{d}T}}$, $\mathcal{D}_U$ be the uniform distribution over $[0,1]^d$, and let $a^1 \in \mathcal{S}$ be arbitrary.
For $t = 1, \ldots, T$:
Play action a^t; Observe loss vector ℓ^t and suffer loss $\langle \ell^t, a^t \rangle$.
Update:

$$a^{t+1} = \operatorname*{argmin}_{a \in \mathcal{S}} \left[\eta \sum_{s \leq t} \langle \ell^s, a \rangle + \langle \xi^t, a \rangle \right]$$

where ξ^t is drawn independently for each t from the distribution $\mathcal{D}_U$.

Theorem 5.2.6 (Kalai and Vempala (2005)). *For any sequence of loss vectors $\ell^1, \ldots, \ell^T$, the FTPL algorithm has regret*

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \ell^t, a^t \rangle \right] - \min_{a \in \mathcal{S}} \sum_{t=1}^{T} \langle \ell^t, a \rangle \leq 2d^{5/4} M \sqrt{T}$$

where the randomness is taken over the perturbations ξ^t across rounds.

5.2.1. Generalization Error

In this section, we observe that the error rate of a classifier D, as well as the degree to which it violates γ-fairness (for both statistical parity and false positive rates) can be accurately approximated with the empirical estimates for these quantities on a dataset (drawn i.i.d. from the underlying distribution $\mathcal{P}$) so long as the dataset is sufficiently large. Once we establish this fact, since our main interest is in the computational problem of auditing and learning, in the rest of the paper, we assume that we have direct access to the underlying distribution (or equivalently, that the empirical data defines the distribution of interest), and do not make further reference to sample complexity or overfitting issues.

A standard VC dimension bound (see, e.g. Kearns and Vazirani (1994b)) states:

Theorem 5.2.7. *Fix a class of functions $\mathcal{H}$. For any distribution $\mathcal{P}$, let $S \sim \mathcal{P}^m$ be a dataset consisting of m examples (X_i, y_i) sampled i.i.d. from $\mathcal{P}$. Then for any $0 < \delta < 1$, with probability $1 - \delta$, for every $h \in \mathcal{H}$, we have:*

$$|err(h, \mathcal{P}) - err(h, S)| \leq O\left(\sqrt{\frac{\text{VCDIM}(\mathcal{H}) \log m + \log(1/\delta)}{m}}\right)$$

where $err(h, S) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[h(X_i) \neq y_i]$.

The above theorem implies that so long as $m \geq \tilde{O}(\text{VCDIM}(\mathcal{H})/\varepsilon^2)$, then minimizing error over the empirical sample S suffices to minimize error up to an additive ε term on the true distribution $\mathcal{P}$. Below, we give two analogous statements for fairness constraints:

Theorem 5.2.8 (SP Uniform Convergence)**.** *Fix a class of functions $\mathcal{H}$ and a class of group indicators $\mathcal{G}$. For any distribution $\mathcal{P}$, let $S \sim \mathcal{P}^m$ be a dataset consisting of m examples (X_i, y_i) sampled i.i.d. from $\mathcal{P}$. Then for any $0 < \delta < 1$, with probability $1 - \delta$, for every $h \in \mathcal{H}$ and $g \in \mathcal{G}$*

$$\left|\alpha_{SP}(g, \mathcal{P}_S)\, \beta_{SP}(g, h, \mathcal{P}_S) - \alpha_{SP}(g, \mathcal{P})\, \beta_{SP}(g, h, \mathcal{P})\right| \leq \tilde{O}\left(\sqrt{\frac{(\text{VCDIM}(\mathcal{H}) + \text{VCDIM}(\mathcal{G})) \log m + \log(1/\delta)}{m}}\right) \quad (5.1)$$

where $\mathcal{P}_S$ denotes the empirical distribution over the realized sample S.

Similarly:

Theorem 5.2.9 (FP Uniform Convergence)**.** *Fix a class of functions $\mathcal{H}$ and a class of group indicators $\mathcal{G}$. For any distribution $\mathcal{P}$, let $S \sim \mathcal{P}^m$ be a dataset consisting of m examples (X_i, y_i) sampled i.i.d. from $\mathcal{P}$. Then for any $0 < \delta < 1$, with probability $1 - \delta$, for every $h \in \mathcal{H}$ and $g \in \mathcal{G}$,*

we have:

$$\left|\alpha_{FP}(g,\mathcal{P})\,\beta_{FP}(g,D,\mathcal{P}) - \alpha_{FP}(g,\mathcal{P})\,\beta_{FP}(g,D,\mathcal{P})\right| \leq \tilde{O}\left(\sqrt{\frac{(\mathrm{VCDIM}(\mathcal{H})+\mathrm{VCDIM}(\mathcal{G}))\log m + \log(1/\delta)}{m}}\right), \quad (5.2)$$

where $\mathcal{P}_S$ denotes the empirical distribution over the realized sample S.

These theorems together imply that for both SP and FP subgroup fairness, the degree to which a group g violates the constraint of γ-fairness can be estimated up to error ε, so long as $m \geq \tilde{O}((\mathrm{VCDIM}(\mathcal{H}) + \mathrm{VCDIM}(\mathcal{G}))/\varepsilon^2)$. The proofs can be found in Appendix A.3.2.

5.3. Equivalence of Auditing and Weak Agnostic Learning

In this section, we give a reduction from the problem of auditing both statistical parity and false positive rate fairness, to the problem of agnostic learning, and vice versa. This has two implications. The main implication is that, from a worst-case analysis point of view, auditing is computationally hard in almost every case (since it inherits this pessimistic state of affairs from agnostic learning). However, worst-case hardness results in learning theory have not prevented the successful practice of machine learning, and there are many heuristic algorithms that in real-world cases successfully solve "hard" agnostic learning problems. Our reductions also imply that these heuristics can be used successfully as auditing algorithms, and we exploit this in the development of our algorithmic results and their experimental evaluation.

We make the following mild assumption on the class of group indicators $\mathcal{G}$, to aid in our reductions. It is satisfied by most natural classes of functions, but is in any case essentially without loss of generality (since learning negated functions can be simulated by learning the original function class on a dataset with flipped class labels).

Assumption 5.3.1. *We assume the set of group indicators $\mathcal{G}$ satisfies closure under negation: for any $g \in \mathcal{G}$, we also have $\neg g \in \mathcal{G}$.*

Recalling that $X = (x, x')$ and the following notions will be useful for describing our results:

- $\mathrm{SP}(D) = \Pr_{\mathcal{P},D}[D(X) = 1]$ and $\mathrm{FP}(D) = \Pr_{D,\mathcal{P}}[D(X) = 1 \mid y = 0]$.
- $\alpha_{SP}(g, \mathcal{P}) = \Pr_{\mathcal{P}}[g(x) = 1]$ and $\alpha_{FP}(g, \mathcal{P}) = \Pr_{\mathcal{P}}[g(x) = 1, y = 0]$.
- $\beta_{SP}(g, D, \mathcal{P}) = |\mathrm{SP}(D) - \mathrm{SP}(D, g)|$ and $\beta_{FP}(g, D, \mathcal{P}) = |\mathrm{FP}(D) - \mathrm{FP}(D, g)|$.
- P^D: the marginal distribution on $(x, D(X))$.
- $P^D_{y=0}$: the conditional distribution on $(x, D(X))$, conditioned on $y = 0$.

We will think about these as the target distributions for a learning problem: i.e. the problem of learning to predict $D(X)$ from only the protected features x. We will relate the ability to agnostically learn on these distributions, to the ability to audit D given access to the original distribution $P_{\text{audit}}(D)$.

5.3.1. Statistical Parity Fairness

We give our reduction first for SP subgroup fairness. The reduction for FP subgroup fairness will follow as a corollary, since auditing for FP subgroup fairness can be viewed as auditing for statistical parity fairness on the subset of the data restricted to $y = 0$.

Theorem 5.3.2. *Fix any distribution $\mathcal{P}$, and any set of group indicators $\mathcal{G}$. Then for any $\gamma, \varepsilon > 0$, the following relationships hold:*

- *If there is a $(\gamma/2, (\gamma/2 - \varepsilon))$ auditing algorithm for $\mathcal{G}$ for all D such that $\mathrm{SP}(D) = 1/2$, then the class $\mathcal{F}$ is $(\gamma, \gamma/2 - \varepsilon)$-weakly agnostically learnable under P^D.*
- *If $\mathcal{G}$ is $(\gamma, \gamma - \varepsilon)$-weakly agnostically learnable under distribution P^D for all D such that $\mathrm{SP}(D) = 1/2$, then there is a $(\gamma, (\gamma - \varepsilon)/2)$ auditing algorithm for $\mathcal{G}$ for SP fairness under $\mathcal{P}$.*

We will prove Theorem 5.3.2 in two steps. First, we show that any unfair certificate f for D has non-trivial error for predicting the decision made by D from the sensitive attributes.

Lemma 5.3.3. *Suppose that the base rate* $\mathrm{SP}(D) \leq 1/2$ *and there exists a function* f *such that*

$$\alpha_{SP}(g, \mathcal{P})\, \beta_{SP}(g, D, \mathcal{P}) = \gamma.$$

Then

$$\max\{\Pr[D(X) = f(x)], \Pr[D(X) = \neg f(x)]\} \geq \mathrm{SP}(D) + \gamma.$$

Proof. To simplify notations, let $b = \mathrm{SP}(D)$ denote the base rate, $\alpha = \alpha_{SP}$ and $\beta = \beta_{SP}$. First, observe that either $\Pr[D(X) = 1 \mid f(x) = 1] = b + \beta$ or $\Pr[D(X) = 1 \mid f(x) = 1] = b - \beta$ holds.

In the first case, we know $\Pr[D(X) = 1 \mid f(x) = 0] < b$, and so $\Pr[D(X) = 0 \mid f(x) = 0] > 1 - b$. It follows that

$$\begin{aligned}
\Pr[D(X) = f(x)] &= \Pr[D(X) = f(x) = 1] + \Pr[D(X) = f(x) = 0] \\
&= \Pr[D(X) = 1 \mid f(x) = 1]\Pr[f(x) = 1] + \Pr[D(X) = 0 \mid f(x) = 0]\Pr[f(x) = 0] \\
&> \alpha(b + \beta) + (1 - \alpha)(1 - b) \\
&= (\alpha - 1)b + (1 - \alpha)(1 - b) + b + \alpha\beta \\
&= (1 - \alpha)(1 - 2b) + b + \alpha\beta.
\end{aligned}$$

In the second case, we have $\Pr[D(X) = 0 \mid f(x) = 1] = (1 - b) + \beta$ and $\Pr[D(X) = 1 \mid f(x) = 0] > b$. We can then bound

$$\begin{aligned}
\Pr[D(X) = f(x)] &= \Pr[D(X) = 1 \mid f(x) = 0]\Pr[f(x) = 0] + \Pr[D(X) = 0 \mid f(x) = 1]\Pr[f(x) = 1] \\
&> (1 - \alpha)b + \alpha(1 - b + \beta) = \alpha(1 - 2b) + b + \alpha\beta.
\end{aligned}$$

In both cases, we have $(1 - 2b) \geq 0$ by our assumption on the base rate. Since $\alpha \in [0, 1]$, we know

$$\max\{\Pr[D(X) = f(x)], \Pr[D(X) = \neg f(x)]\} \geq b + \alpha\beta = b + \gamma$$

which recovers our bound. □

In the next step, we show that if there exists any function f that accurately predicts the decisions made by the algorithm D, then either f or $\neg f$ can serve as an unfairness certificate for D.

Lemma 5.3.4. *Suppose that the base rate* $\mathrm{SP}(D) \geq 1/2$ *and there exists a function* f *such that* $\Pr[D(X) = f(x)] \geq \mathrm{SP}(D) + \gamma$ *for some value* $\gamma \in (0, 1/2)$*. Then there exists a function* g *such that*

$$\alpha_{SP}(g, \mathcal{P})\, \beta_{SP}(g, D, \mathcal{P}) \geq \gamma/2,$$

where $g \in \{f, \neg f\}$.

Proof. Let $b = \mathrm{SP}(D)$. We can expand $\Pr[D(X) = f(x)]$ as follows:

$$\begin{aligned}\Pr[D(X) = f(x)] &= \Pr[D(X) = f(x) = 1] + \Pr[D(X) = f(x) = 0] \\ &= \Pr[D(X) = 1 \mid f(x) = 1]\Pr[f(x) = 1] + \Pr[D(X) = 0 \mid f(x) = 0]\Pr[f(x) = 0]\end{aligned}$$

This means

$$\begin{aligned}&\Pr[D(X) = f(x)] - b \\ &= (\Pr[D(X) = 1 \mid f(x) = 1] - b)\Pr[f(x) = 1] + (\Pr[D(X) = 0 \mid f(x) = 0] - b)\Pr[f(x) = 0] \geq \gamma\end{aligned}$$

Suppose that $(\Pr[D(X) = 1 \mid f(x) = 1] - b)\Pr[f(x) = 1] \geq \gamma/2$, then our claim holds with $g = f$. Suppose not, then we must have

$$(\Pr[D(X) = 0 \mid f(x) = 0] - b)\Pr[f(x) = 0] = ((1 - b) - \Pr[D(X) = 1 \mid f(x) = 0])\Pr[f(x) = 0] \geq \gamma/2$$

Note that by our assumption $b \geq (1 - b)$. This means

$$(b - \Pr[D(X) = 1 \mid f(x) = 0])\Pr[f(x) = 0] \geq ((1 - b) - \Pr[D(X) = 1 \mid f(x) = 0])\Pr[f(x) = 0] \geq \gamma/2$$

which implies that our claim holds with $g = \neg f$. □

Proof of Theorem 5.3.2. Suppose that the class $\mathcal{G}$ satisfies $\min_{f \in \mathcal{G}} err(f, P^D) \leq 1/2 - \gamma$. Then by Lemma 5.3.4, there exists some $g \in \mathcal{G}$ such that $\Pr[g(x) = 1]|\Pr[D(X) = 1 \mid g(x) = 1] - \mathrm{SP}(D)| \geq \gamma/2$. By the assumption of auditability, we can then use the auditing algorithm to find a group $g' \in \mathcal{G}$ that is an $(\gamma/2 - \varepsilon)$-unfair certificate of D. By Lemma 5.3.3, we know that either g' or $\neg g'$ predicts D with an accuracy of at least $1/2 + (\gamma/2 - \varepsilon)$.

In the reverse direction, consider the auditing problem on the classifier D. We can treat each pair $(x, D(X))$ as a labelled example and learn a hypothesis in $\mathcal{G}$ that approximates the decisions made by D. Suppose that D is γ-unfair. Then by Lemma 5.3.3, we know that there exists some $g \in \mathcal{G}$ such that $\Pr[D(X) = g(x)] \geq 1/2 + \gamma$. Therefore, the weak agnostic learning algorithm from the hypothesis of the theorem will return some g' with $\Pr[D(X) = g'(x)] \geq 1/2 + (\gamma - \varepsilon)$. By Lemma 5.3.4, we know g' or $\neg g'$ is a $(\gamma - \varepsilon)/2$-unfair certificate for D. □

5.3.2. False Positive Fairness

A corollary of the above reduction is an analogous equivalence between auditing for FP subgroup fairness and agnostic learning. This is because a FP fairness constraint can be viewed as a statistical parity fairness constraint on the subset of the data such that $y = 0$. Therefore, Theorem 5.3.2 implies the following:

Corollary 5.3.5. *Fix any distribution $\mathcal{P}$, and any set of group indicators $\mathcal{G}$. The following two relationships hold:*

- *If there is a $(\gamma/2, (\gamma/2 - \varepsilon))$ auditing algorithm for $\mathcal{F}$ for all D such that $\mathrm{FP}(D) = 1/2$, then the class $\mathcal{G}$ is $(\gamma, \gamma/2 - \varepsilon)$-weakly agnostically learnable under $P^D_{y=0}$.*
- *If $\mathcal{G}$ is $(\gamma, \gamma - \varepsilon)$–weakly agnostically learnable under distribution $P^D_{y=0}$ for all D such that $\mathrm{FP}(D) = 1/2$, then there is a $(\gamma, (\gamma - \varepsilon)/2)$ auditing algorithm for FP subgroup fairness for $\mathcal{G}$ under distribution $\mathcal{P}$.*

5.3.3. *Worst-Case Intractability of Auditing*

While we shall see in subsequent sections that the equivalence given above has positive algorithmic and experimental consequences, from a purely theoretical perspective the reduction of agnostic learning to auditing has strong negative worst-case implications. More precisely, we can import a long sequence of formal intractability results for agnostic learning to obtain:

Theorem 5.3.6. *Under standard complexity-theoretic intractability assumptions, for $\mathcal{G}$ the classes of conjunctions of boolean attributes, linear threshold functions, or bounded-degree polynomial threshold functions, there exist distributions P such that the auditing problem cannot be solved in polynomial time, for either statistical parity or false positive fairness.*

The proof of this theorem follows from Theorem 5.3.2, Corollary 5.3.5, and the following negative results from the learning theory literature. Feldman et al. (2012a) show a strong negative result for weak agnostic learning for conjunctions: given a distribution on labeled examples from the hypercube such that there exists a monomial (or conjunction) consistent with $(1-\varepsilon)$-fraction of the examples, it is NP-hard to find a halfspace that is correct on $(1/2+\varepsilon)$-fraction of the examples, for arbitrary constant $\varepsilon > 0$. Diakonikolas et al. (2011b) show that under the Unique Games Conjecture, no polynomial-time algorithm can find a degree-d polynomial threshold function (PTF) that is consistent with $(1/2+\varepsilon)$ fraction of a given set of labeled examples, even if there exists a degree-d PTF that is consistent with a $(1-\varepsilon)$ fraction of the examples. Diakonikolas et al. (2011b) also show that it is NP-Hard to find a degree-2 PTF that is consistent with a $(1/2+\varepsilon)$ fraction of a given set of labeled examples, even if there exists a halfspace (degree-1 PTF) that is consistent with a $(1-\varepsilon)$ fraction of the examples.

While Theorem 5.3.6 shows that certain natural subgroup classes $\mathcal{G}$ yield intractable auditing problems in the worst case, in the rest of the paper we demonstrate that effective heuristics for this problem on specific (non-worst case) distributions can be used to derive an effective and practical learning algorithm for subgroup fairness.

5.4. A Learning Algorithm Subject to Fairness Constraints $\mathcal{G}$

In this section, we present an algorithm for training a (randomized) classifier that satisfies false-positive subgroup fairness simultaneously for all protected subgroups specified by a family of group indicator functions $\mathcal{G}$. All of our techniques also apply to a statistical parity or false negative rate constraint.

Let S denote a set of n labeled examples $\{z_i = (x_i, x_i'), y_i)\}_{i=1}^n$, and let $\mathcal{P}$ denote the empirical distribution over this set of examples. Let $\mathcal{H}$ be a hypothesis class defined over both the protected and unprotected attributes, and let $\mathcal{G}$ be a collection of group indicators over the protected attributes. We assume that $\mathcal{H}$ contains a constant classifier (which implies that there is at least one fair classifier to be found, for any distribution).

Our goal will be to find the distribution over classifiers from $\mathcal{H}$ that minimizes classification error subject to the fairness constraint over $\mathcal{G}$. We will design an iterative algorithm that, when given access to a CSC oracle, computes an optimal randomized classifier in polynomial time.

Let D denote a probability distribution over $\mathcal{H}$. Consider the following *Fair ERM (Empirical Risk Minimization)* problem:

$$\min_{D \in \Delta_{\mathcal{H}}} \mathop{\mathbb{E}}_{h \sim D} [err(h, \mathcal{P})] \tag{5.3}$$

$$\text{such that } \forall g \in \mathcal{G} \qquad \alpha_{FP}(g, \mathcal{P})\, \beta_{FP}(g, D, \mathcal{P}) \leq \gamma. \tag{5.4}$$

where $err(h, \mathcal{P}) = \Pr_{\mathcal{P}}[h(x, x') \neq y]$, and the quantities α_{FP} and β_{FP} are defined in Definition 5.2.3. We will write OPT to denote the objective value at the optimum for the Fair ERM problem, that is the minimum error achieved by a γ-fair distribution over the class $\mathcal{H}$.

Observe that the optimization is feasible for any distribution $\mathcal{P}$: the constant classifiers that labels all points 1 or 0 satisfy all subgroup fairness constraints. At the moment, the number

of decision variables and constraints may be infinite (if $\mathcal{H}$ and $\mathcal{G}$ are infinite hypothesis classes), but we will address this momentarily.

Assumption 5.4.1 (Cost-Sensitive Classification Oracle)**.** *We assume our algorithm has access to the cost-sensitive classication oracles* CSC($\mathcal{H}$) *and* CSC($\mathcal{G}$) *over the classes $\mathcal{H}$ and $\mathcal{G}$.*

Our main theoretical result is an computationally efficient oracle-based algorithm for solving the Fair ERM problem.

Theorem 5.4.2. *Fix any $\nu, \delta \in (0,1)$. Then given an input of n data points and accuracy parameters ν, δ and access to oracles* CSC($\mathcal{H}$) *and* CSC($\mathcal{G}$)*, there exists an algorithm runs in polynomial time, and with probability at least $1-\delta$, output a randomized classifier $\hat{D}$ such that $err(\hat{D}, \mathcal{P}) \leq \text{OPT} + \nu$, and for any $g \in \mathcal{G}$, the fairness constraint violations satisfies*

$$\alpha_{FP}(g, \mathcal{P})\, \beta_{FP}(g, \hat{D}, \mathcal{P}) \leq \gamma + O(\nu).$$

Overview of our solution. We present our solution in steps:

- **Step 1: Fair ERM as LP.** First, we rewrite the Fair ERM problem as a linear program with finitely many decision variables and constraints even when $\mathcal{H}$ and $\mathcal{G}$ are infinite. To do this, we take advantage of the fact that Sauer's Lemma lets us bound the number of labellings that any hypothesis class $\mathcal{H}$ of bounded VC dimension can induce on any fixed dataset. The LP has one variable for each of these possible labellings, rather than one variable for each hypothesis. Moreover, again by Sauer's Lemma, we have one constraint for each of the finitely many possible subgroups induced by $\mathcal{G}$ on the fixed dataset, rather than one for each of the (possibly infinitely many) subgroups definable over arbitrary datasets. This step is important — it will guarantee that strong duality holds.

- **Step 2: Formulation as Game.** We then derive the partial Lagrangian of the LP, and note that computing an approximately optimal solution to this LP is equivalent to finding an approximate minmax solution for a corresponding zero-sum game, in

which the payoff function U is the value of the Lagrangian. The pure strategies of the primal or "Learner" player correspond to classifiers $h \in \mathcal{H}$, and the pure strategies of the dual or "Auditor" player correspond to subgroups $g \in \mathcal{G}$. Intuitively, the Learner is trying to minimize the sum of the prediction error and a fairness penalty term (given by the Lagrangian), and the Auditor is trying to penalize the fairness violation of the Learner by first identifying the subgroup with the greatest fairness violation and putting all the weight on the dual variable corresponding to this subgroup. In order to reason about convergence, we restrict the set of dual variables to lie in a bounded set: C times the probability simplex. C is a parameter that we have to set in the proof of our theorem to give the best theoretical guarantees — but it is also a parameter that we will vary in the experimental section.

- **Step 3: Best Responses as CSC.** We observe that given a mixed strategy for the Auditor, the best response problem of the Learner corresponds to a CSC problem. Similarly, given a mixed strategy for the Learner, the best response problem of the Auditor corresponds to an auditing problem (which can be represented as a CSC problem). Hence, if we have oracles for solving CSC problems, we can compute best responses for both players, in response to arbitrary mixed strategies of their opponents.

- **Step 4: FTPL for No-Regret.** Finally, we show that the ability to compute best responses for each player is sufficient to implement dynamics known to converge quickly to equilibrium in zero-sum games. Our algorithm has the Learner play *Follow the Perturbed Leader (FTPL)* Kalai and Vempala (2005), which is a no-regret algorithm, against an Auditor who at every round best responds to the learner's mixed strategy. By the seminal result of Freund and Schapire (1996a), the average plays of both players converge to an approximate equilibrium. In order to implement this in polynomial time, we need to represent the loss of the learner as a low-dimensional linear optimization problem. To do so, we first define an appropriately translated CSC

problem for any mixed strategy λ by the Auditor, and cast it as a linear optimization problem.

5.4.1. *Rewriting the Fair ERM Problem*

To rewrite the Fair ERM problem, we note that even though both $\mathcal{G}$ and $\mathcal{H}$ can be infinite sets, the sets of possible labellings on the data set S induced by these classes are finite. More formally, we will write $\mathcal{G}(S)$ and $\mathcal{H}(S)$ to denote the set of all labellings on S that are induced by $\mathcal{G}$ and $\mathcal{H}$ respectively, that is

$$\mathcal{G}(S) = \{(g(x_1), \ldots, g(x_n)) \mid g \in \mathcal{G}\} \qquad \text{and,} \qquad \mathcal{H}(S) = \{(h(X_1), \ldots, h(X_n)) \mid h \in \mathcal{H}\}$$

We can bound the cardinalities of $\mathcal{G}(S)$ and $\mathcal{H}(S)$ using Sauer's Lemma.

Lemma 5.4.3 (Sauer's Lemma (see e.g. Kearns and Vazirani (1994b))). *Let S be a data set of size n. Let $d_1 = \text{VCDIM}(\mathcal{H})$ and $d_2 = \text{VCDIM}(\mathcal{G})$ be the VC-dimensions of the two classes. Then*

$$|\mathcal{H}(S)| \leq O\left(n^{d_1}\right) \qquad \textit{and} \qquad |\mathcal{G}(S)| \leq O\left(n^{d_2}\right).$$

Given this observation, we can then consider an equivalent optimization problem where the distribution D is over the set of labellings in $\mathcal{H}(S)$, and the set of subgroups are defined by the labellings in $\mathcal{G}(S)$. We will view each g in $\mathcal{G}(S)$ as a Boolean function.

To simplify notations, we will define the following "fairness violation" functions for any $g \in \mathcal{G}$ and any $h \in \mathcal{H}$:

$$\Phi_+(h, g) \equiv \alpha_{FP}(g, P)\,(\text{FP}(h) - \text{FP}(h, g)) - \gamma \tag{5.5}$$

$$\Phi_-(h, g) \equiv \alpha_{FP}(g, \mathcal{P})\,(\text{FP}(h, g) - \text{FP}(h)) - \gamma \tag{5.6}$$

Moreover, for any distribution D over $\mathcal{H}$, for any sign $\bullet \in \{+,-\}$

$$\Phi_\bullet(D,g) = \mathop{\mathbb{E}}_{h\sim D}[\Phi_\bullet(h,g)].$$

Claim 5.4.4. *For any $g \in \mathcal{G}$, $h \in \mathcal{H}$, and any $\nu > 0$,*

$$\max\{\Phi_+(D,g),\Phi_-(D,g)\} \le \nu \quad \textit{if and only if} \quad \alpha_{FP}(g,\mathcal{P})\,\beta_{FP}(g,D,\mathcal{P}) \le \gamma + \nu.$$

Thus, we will focus on the following equivalent optimization problem.

$$\min_{D\in\Delta_{\mathcal{H}(S)}} \mathop{\mathbb{E}}_{h\sim D}[err(h,\mathcal{P})] \tag{5.7}$$

$$\text{such that for each } g \in \mathcal{G}(S): \qquad \Phi_+(D,g) \le 0 \tag{5.8}$$

$$\Phi_-(D,g) \le 0 \tag{5.9}$$

For each pair of constraints (5.8) and (5.9), corresponding to a group $g \in \mathcal{G}(S)$, we introduce a pair of dual variables λ_g^+ and λ_g^-. The partial Lagrangian of the linear program is the following:

$$\mathcal{L}(D,\lambda) = \mathop{\mathbb{E}}_{h\sim D}[err(h,\mathcal{P})] + \sum_{g\in\mathcal{G}(S)} \left(\lambda_g^+\Phi_+(D,g) + \lambda_g^-\Phi_-(D,g)\right)$$

By Sion's minmax theorem (Sion, 1958), we have

$$\min_{D\in\Delta_{\mathcal{H}(S)}} \max_{\lambda\in\mathbb{R}_+^{2|\mathcal{G}(S)|}} \mathcal{L}(p,\lambda) = \max_{\lambda\in\mathbb{R}_+^{2|\mathcal{G}(S)|}} \min_{D\in\Delta_{\mathcal{H}(S)}} \mathcal{L}(p,\lambda) = \mathrm{OPT}$$

where OPT denotes the optimal objective value in the fair ERM problem. Similarly, the distribution $\arg\min_D \max_\lambda \mathcal{L}(D,\lambda)$ corresponds to an optimal feasible solution to the fair ERM linear program. Thus, finding an optimal solution for the fair ERM problem reduces

to computing a minmax solution for the Lagrangian. Our algorithms will both compute such a minmax solution by iteratively optimizing over both the primal variables D and dual variables λ. In order to guarantee convergence in our optimization, we will restrict the dual space to the following bounded set:

$$\Lambda = \{\lambda \in \mathbb{R}_+^{2|\mathcal{G}(S)|} \mid \|\lambda\|_1 \leq C\}.$$

where C will be a parameter of our algorithm. Since Λ is a compact and convex set, the minmax condition continues to hold (Sion, 1958):

$$\min_{D \in \Delta_{\mathcal{H}(S)}} \max_{\lambda \in \Lambda} \mathcal{L}(D, \lambda) = \max_{\lambda \in \Lambda} \min_{D \in \Delta_{\mathcal{H}(S)}} \mathcal{L}(D, \lambda) \tag{5.10}$$

If we knew an upper bound C on the ℓ_1 norm of the optimal dual solution, then this restriction on the dual solution would not change the minmax solution of the program. We do not in general know such a bound. However, we can show that even though we restrict the dual variables to lie in a bounded set, any approximate minmax solution to Equation (5.10) is also an approximately optimal and approximately feasible solution to the original fair ERM problem.

Theorem 5.4.5. *Let $(\hat{D}, \hat{\lambda})$ be a ν-approximate minmax solution to the Λ-bounded Lagrangian problem in the sense that*

$$\mathcal{L}(\hat{D}, \hat{\lambda}) \leq \min_{D \in \Delta_{\mathcal{H}(S)}} \mathcal{L}(D, \hat{\lambda}) + \nu \quad \textit{and}, \quad \mathcal{L}(\hat{D}, \hat{\lambda}) \geq \max_{\lambda \in \Lambda} \mathcal{L}(\hat{D}, \lambda) - \nu.$$

Then $err(\hat{D}, \mathcal{P}) \leq \mathrm{OPT} + 2\nu$ and for any $g \in \mathcal{G}(S)$,

$$\alpha_{FP}(g, \mathcal{P})\, \beta_{FP}(g, \hat{D}, \mathcal{P}) \leq \gamma + \frac{1 + 2\nu}{C}.$$

5.4.2. Zero-Sum Game Formulation

To compute an approximate minmax solution, we will first view Equation (5.10) as the following two player zero-sum matrix game. The Learner (or the minimization player) has pure strategies corresponding to $\mathcal{H}$, and the Auditor (or the maximization player) has pure strategies corresponding to the set of vertices Λ_{pure} in Λ — more precisely, each vertex or pure strategy either is the all zero vector or consists of a choice of a $g \in \mathcal{G}(S)$, along with the sign + or − that the corresponding g-fairness constraint will have in the Lagrangian. More formally, we write

$$\Lambda_{\text{pure}} = \{\lambda \in \Lambda \text{ with } \lambda_g^{\bullet} = C \mid g \in \mathcal{G}(S), \bullet \in \{\pm\}\} \cup \{\mathbf{0}\}$$

Even though the number of pure strategies scales linearly with $|\mathcal{G}(S)|$, our algorithm will never need to actually represent such vectors explicitly. Note that any vector in Λ can be written as a convex combination of the maximization player's pure strategies, or in other words: as a mixed strategy for the Auditor. For any pair of actions $(h, \lambda) \in \mathcal{H} \times \Lambda_{\text{pure}}$, the payoff is defined as

$$U(h, \lambda) = err(h, \mathcal{P}) + \sum_{g \in \mathcal{G}(S)} \left(\lambda_g^{+}\Phi_{+}(h, g) + \lambda_g^{-}\Phi_{-}(h, g)\right).$$

Claim 5.4.6. *Let $D \in \Delta_{\mathcal{H}(S)}$ and $\lambda \in \Lambda$ such that (p, λ) is a ν-approximate minmax equilibrium in the zero-sum game defined above. Then (p, λ) is also a ν-approximate minmax solution for Equation (5.10).*

Our problem reduces to finding an approximate equilibrium for this game. A key step in our solution is the ability to compute best responses for both players in the game, which we now show can be solved by the cost-sensitive classication (CSC) oracles.

Learner's best response as CSC. Fix any mixed strategy (dual solution) $\lambda \in \Lambda$ of the Auditor. The Learner's best response is given by:

$$\operatorname*{argmin}_{D \in \Delta_{\mathcal{H}(S)}} err(h, \mathcal{P}) + \sum_{g \in \mathcal{G}(S)} \left(\lambda_g^+ \Phi_+(D, g) + \lambda_g^- \Phi_-(D, g) \right) \tag{5.11}$$

Note that it suffices for the Learner to optimize over deterministic classifiers $h \in \mathcal{H}$, rather than distributions over classifiers. This is because the Learner is solving a linear optimization problem over the simplex, and so always has an optimal solution at a vertex (i.e. a single classifier $h \in \mathcal{H}$). We can reduce this problem to one that can be solved with a single call to a CSC oracle. In particular, we can assign costs to each example (X_i, y_i) as follows:

- if $y_i = 1$, then $c_i^0 = 0$ and $c_i^1 = -\frac{1}{n}$;
- otherwise, $c_i^0 = 0$ and

$$c_i^1 = \frac{1}{n} + \frac{1}{n} \sum_{g \in \mathcal{G}(S)} (\lambda_g^+ - \lambda_g^-) \left(\Pr[g(x) = 1 \mid y = 0] - \mathbf{1}[g(x_i) = 1] \right) \tag{5.12}$$

Given a fixed set of dual variables λ, we will write $\mathrm{LC}(\lambda) \in \mathbb{R}^n$ to denote the vector of costs for labelling each datapoint as 1. That is, $\mathrm{LC}(\lambda)$ is the vector such that for any $i \in [n]$, $\mathrm{LC}(\lambda)_i = c_i^1$.

Remark 5.4.7. *Note that in defining the costs above, we have translated them from their most natural values so that the cost of labeling any example with 0 is 0. In doing so, we recall that by Claim 2.2.2, the solution to a cost-sensitive classification problem is invariant to translation. As we will see, this will allow us to formulate the learner's optimization problem as a low-dimensional linear optimization problem, which will be important for an efficient implementation of follow the perturbed leader. In particular, if we find a hypothesis that produces the n labels $y = (y_1, \ldots, y_n)$ for the n points in our dataset, then the cost of this labelling in the CSC problem is by construction $\langle \mathrm{LC}(\lambda), y \rangle$.*

Auditor's best response as CSC. Fix any mixed strategy (primal solution) $p \in \Delta_{\mathcal{H}(S)}$ of the Learner. The Auditor's best response is given by:

$$\operatorname*{argmax}_{\lambda \in \Lambda} \; err(D, \mathcal{P}) + \sum_{g \in \mathcal{G}(S)} \left(\lambda_g^+ \Phi_+(D,g) + \lambda_g^- \Phi_-(D,g) \right) = \operatorname*{argmax}_{\lambda \in \Lambda} \sum_{g \in \mathcal{G}(S)} \left(\lambda_g^+ \Phi_+(D,g) + \lambda_g^- \Phi_-(D,g) \right) \tag{5.13}$$

To find the best response, consider the problem of computing $(\hat{g}, \hat{\bullet}) = \operatorname{argmax}_{(g,\bullet)} \Phi_\bullet(D,g)$. There are two cases. In the first case, p is a strictly feasible primal solution: that is $\Phi_{\hat{\bullet}}(D,\hat{g}) < 0$. In this case, the solution to (5.13) sets $\lambda = \mathbf{0}$. Otherwise, if p is not strictly feasible, then by the following Lemma 5.4.8 the best response is to set $\lambda_{\hat{g}}^{\hat{\bullet}} = C$ (and all other coordinates to 0).

Lemma 5.4.8. *Fix any $\overline{D} \in \Delta_{\mathcal{H}(S)}$ such that that $\max_{g \in \mathcal{G}(S)} \{\Phi_+(\overline{D},g), \Phi_-(\overline{D},g)\} > 0$. Let $\lambda' \in \Lambda$ be vector with one non-zero coordinate $(\lambda')_{g'}^{\bullet'} = C$, where*

$$(g', \bullet') = \operatorname*{argmax}_{(g,\bullet) \in \mathcal{G}(S) \times \{\pm\}} \{\Phi_\bullet(\overline{D}, g)\}$$

Then $\mathcal{L}(\overline{D}, \lambda') \geq \max_{\lambda \in \Lambda} \mathcal{L}(\overline{D}, \lambda)$.

Therefore, it suffices to solve for $\operatorname{argmax}_{(g,\bullet)} \Phi_\bullet(D,g)$. We proceed by solving $\operatorname{argmax}_g \Phi_+(D,g)$ and $\operatorname{argmax}_g \Phi_-(D,g)$ separately: both problems can be reduced to a cost-sensitive classification problem. To solve for $\operatorname{argmax}_g \Phi_+(D,g)$ with a CSC oracle, we assign costs to each example (X_i, y_i) as follows:

- if $y_i = 1$, then $c_i^0 = 0$ and $c_i^1 = 0$;
- otherwise, $c_i^0 = 0$ and

$$c_i^1 = \frac{-1}{n} \left[\mathop{\mathbb{E}}_{h \sim D} [\mathrm{FP}(h)] - \mathop{\mathbb{E}}_{h \sim D} [h(X_i)] \right] \tag{5.14}$$

To solve for $\text{argmax}_g \Phi_-(D,g)$ with a CSC oracle, we assign the same costs to each example (X_i, y_i), except when $y_i = 0$, labeling "1" incurs a cost of

$$c_i^1 = \frac{-1}{n}\left[\mathop{\mathbb{E}}_{h\sim D}[h(X_i)] - \mathop{\mathbb{E}}_{h\sim D}[\text{FP}(h)]\right]$$

5.4.3. *Solving the Game with No-Regret Dynamics*

To compute an approximate equilibrium of the zero-sum game, we will simulate the following *no-regret dynamics* between the Learner and the Auditor over rounds: over each of the T rounds, the Learner plays a distribution over the hypothesis class according to a *no-regret* learning algorithm (Follow the Perturbed Leader), and the Auditor plays an approximate best response against the Learner's distribution for that round. By the result of Freund and Schapire (1996a), the average plays of both players over time converge to an approximate equilibrium of the game, as long as the Learner has low regret.

Theorem 5.4.9 (Freund and Schapire (1996a))**.** *Let $D^1, D^2, \ldots, D^T \in \Delta_{\mathcal{H}(S)}$ be a sequence of distributions played by the Learner, and let $\lambda^1, \lambda^2, \ldots, \lambda^T \in \Lambda_{pure}$ be the Auditor's sequence of approximate best responses against these distributions respectively. Let $\overline{D} = \frac{1}{T}\sum_{t=1}^T D^t$ and $\overline{\lambda} = \frac{1}{T}\sum_{t=1}^T \lambda^t$ be the two players' empirical distributions over their strategies. Suppose that the regret of the Learner satisfies*

$$\sum_{t=1}^T \mathop{\mathbb{E}}_{h\sim D^t}\left[U(h,\lambda^t)\right] - \min_{h\in\mathcal{H}(S)} \sum_{t=1}^T U(h,\lambda^t) \le \gamma_L T \quad \textit{and}$$

$$\max_{\lambda\in\Lambda} \sum_{t=1}^T \mathop{\mathbb{E}}_{h\sim D^t}[U(h,\lambda)] - \sum_{t=1}^T \mathop{\mathbb{E}}_{h\sim D^t}\left[U(h,\lambda^t)\right] \le \gamma_A T. \quad (5.15)$$

Then $(\overline{D}, \overline{\lambda})$ is an $(\gamma_L + \gamma_A)$-approximate minimax equilibrium of the game.

Our Learner will play using the Follow the Perturbed Leader (FTPL), which gives a no-regret guarantee. In order to implement FPTL, we will first need to formulate the Learner's best response problem as a linear optimization problem over a low dimensional space. For

each round t, let $\overline{\lambda}^t = \sum_{s<t} \lambda^s$ be the vector representing the sum of the actions played by the auditor over previous rounds, and recall that $\mathrm{LC}(\overline{\lambda}^t)$ is the cost vector given by our cost-sensitive classification reduction. Then the Learner's best response problem against $\overline{\lambda}^t$ is the following linear optimization problem

$$\min_{h \in \mathcal{H}(S)} \langle \mathrm{LC}(\overline{\lambda}^t), h \rangle.$$

To run the FTPL algorithm, the Learner will optimize a "perturbed" version of the problem above. In particular, the Learner will play a distribution D^t over the hypothesis class $\mathcal{H}(S)$ that is implicitely defined by the following sampling operation. To sample a hypothesis h from D^t, the learner solves the following randomized optimization problem:

$$\min_{h \in \mathcal{H}(S)} \langle \mathrm{LC}(\overline{\lambda}^t), h \rangle + \frac{1}{\eta} \langle \xi, h \rangle, \tag{5.16}$$

where η is a parameter and ξ is a noise vector drawn from the uniform distribution over $[0,1]^n$. Note that while it is intractable to explicitly represent the distribution D^t (which has support size scaling with $|\mathcal{H}(S)|$), we can sample from D^t efficiently given access to a cost-sensitive classification oracle for $\mathcal{H}$. By instantiating the standard regret bound of FTPL for online linear optimization (Theorem 5.2.6), we get the following regret bound for the Learner.

Lemma 5.4.10. *Let T be the time horizon for the no-regret dynamics. Let $D^1, \ldots, D^T$ be the sequence of distributions maintained by the Learner's FTPL algorithm with $\eta = \frac{n}{(1+C)} \sqrt{\frac{1}{\sqrt{n}T}}$, and $\lambda^1, \ldots, \lambda^T$ be the sequence of plays by the Auditor. Then*

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{h \sim D^t} \left[U(h, \lambda^t) \right] - \min_{h \in \mathcal{H}(S)} \sum_{t=1}^{T} U(h, \lambda^t) \leq 2n^{1/4}(1+C)\sqrt{T}$$

Now we consider how the Auditor (approximately) best responds to the distribution D^t. The main obstacle is that we do not have an explicit representation for D^t. Thus, our first step is to approximate D^t with an explicitly represented sparse distribution $\hat{D}^t$. We do

that by drawing m i.i.d. samples from D^t, and taking the empirical distribution $\hat{D}^t$ over the sample. The Auditor will best respond to this empirical distribution $\hat{D}^t$. To show that any best response to $\hat{D}^t$ is also an approximate best response to D^t, we will rely on the following uniform convergence lemma, which bounds the difference in expected payoff for any strategy of the auditor, when played against D^t as compared to $\hat{D}^t$.

Lemma 5.4.11. *Fix any $\xi, \delta \in (0,1)$ and any distribution D over $\mathcal{H}(S)$. Let $h^1, \ldots, h^m$ be m i.i.d. draws from p, and $\hat{D}$ be the empirical distribution over the realized sample. Then with probability at least $1-\delta$ over the random draws of h^j's, the following holds,*

$$\max_{\lambda \in \Lambda} \left| \mathop{\mathbb{E}}_{h \sim \hat{D}} [U(h, \lambda)] - \mathop{\mathbb{E}}_{h \sim D} [U(h, \lambda)] \right| \leq \xi,$$

as long as $m \geq c_0 \frac{C^2(\ln(1/\delta) + d_2 \ln(n))}{\xi^2}$ for some absolute constant c_0 and $d_2 = \text{VCDIM}(\mathcal{G})$.

Using Lemma 5.4.11, we can derive a regret bound for the Auditor in the no-regret dynamics.

Lemma 5.4.12. *Let T be the time horizon for the no-regret dynamics. Let $D^1, \ldots, D^T$ be the sequence of distributions maintained by the Learner's FTPL algorithm. For each D^t, let $\hat{D}^t$ be the empirical distribution over m i.i.d. draws from D^t. Let $\lambda^1, \ldots, \lambda^T$ be the Auditor's best responses against $\hat{D}^1, \ldots, \hat{D}^T$. Then with probability $1-\delta$,*

$$\max_{\lambda \in \Lambda} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{h \sim D^t} [U(h, \lambda)] - \sum_{t=1}^{T} \mathop{\mathbb{E}}_{h \sim D^t} \left[U(h, \lambda^t) \right] \leq T \sqrt{\frac{c_0 C^2 (\ln(T/\delta) + d_2 \ln(n))}{m}}$$

for some absolute constant c_0 and $d_2 = \text{VCDIM}(\mathcal{G})$.

Finally, let $\overline{D}$ and $\overline{\lambda}$ be the average of the strategies played by the two players over the course of the dynamics. Note that $\overline{D}$ is an average of many *distributions* with large support, and so $\overline{D}$ itself has support size that is too large to represent explicitely. Thus, we will again approximate $\overline{D}$ with a sparse distribution $\hat{D}$ estimated from a sample drawn from $\overline{D}$. Note that we can efficiently *sample* from $\overline{D}$ given access to a CSC oracle. To sample, we first uniformly randomly select a round $t \in [T]$, and then use the CSC oracle to solve

the sampling problem defined in (5.16), with the noise random variable ξ freshly sampled from its distribution. The full algorithm is described in Algorithm 6 and we present the proof for Theorem 5.4.2 below.

Algorithm 6 FairNR: Fair No-Regret Dynamics

Input: distribution $\mathcal{P}$ over n labelled data points, CSC oracles CSC($\mathcal{H}$) and CSC($\mathcal{G}$), dual bound C, and target accuracy parameter ν, δ
Initialize: Let $C = 1/\nu$, $\overline{\lambda}^0 = \mathbf{0}$, $\eta = \frac{n}{(1+C)}\sqrt{\frac{1}{\sqrt{n}T}}$,

$$m = \frac{(\ln(2T/\delta)d_2\ln(n))C^2c_0T}{\sqrt{n}(1+C)^2\ln(2/\delta)} \quad \text{and,} \quad T = \frac{4\sqrt{n}\ln(2/\delta)}{\nu^4}$$

For $t = 1,\ldots,T$:
Sample from the Learner's FTPL distribution:
For $s = 1,\ldots m$:
[3]Draw a random vector ξ^s uniformly at random from $[0,1]^n$
Use the oracle CSC($\mathcal{H}$) to compute $h^{(s,t)} = \text{argmin}_{h\in\mathcal{H}(S)}\langle \text{LC}(\overline{\lambda}^{(t-1)}), h\rangle + \frac{1}{\eta}\langle \xi^s, h\rangle$
Let $\hat{D}^t$ be the empirical distribution over $\{h^{s,t}\}$

Auditor best responds to $\hat{D}^t$:
Use the oracle CSC($\mathcal{G}$) to compute $\lambda^t = \text{argmax}_\lambda \mathbb{E}_{h\sim\hat{D}}[U(h,\lambda)]$
Update: Let $\overline{\lambda}^t = \sum_{t'\le t}\lambda^{t'}$
Sample from the average distribution $\overline{D} = \sum_{t=1}^T D^t$:
For $s = 1,\ldots m$:
Draw a random number $r \in [T]$ and a random vector ξ^s uniformly at random from $[0,1]^n$
Use the oracle CSC($\mathcal{H}$) to compute $h^{(r,t)} = \text{argmin}_{h\in\mathcal{H}(S)}\langle \text{LC}(\overline{\lambda}^{(r-1)}), h\rangle + \frac{1}{\eta}\langle \xi^s, h\rangle$
Let $\hat{D}$ be the empirical distribution over $\{h^{r,t}\}$
Output: $\hat{D}$ as a randomized classifier

Proof of Theorem 5.4.2. By Theorem 5.4.5, it suffices to show that with probability at least $1-\delta$, $(\hat{D}, \overline{\lambda})$ is a ν-approximate equilibrium in the zero-sum game. As a first step, we will rely on Theorem 5.4.9 to show that $(\overline{D}, \overline{\lambda})$ forms an approximate equilibrium.

By Lemma 5.4.10, the regret of the sequence $D^1,\ldots,D^T$ is bounded by:

$$\gamma_L = \frac{1}{T}\left[\sum_{t=1}^T \mathop{\mathbb{E}}_{h\sim D^t}\left[U(h,\lambda^t)\right] - \min_{h\in\mathcal{H}(S)}\sum_{t=1}^T U(h,\lambda^t)\right] \le \frac{2n^{1/4}(1+C)}{\sqrt{T}}$$

By Lemma 5.4.12, with probability $1 - \delta/2$, we have

$$\gamma_A \leq \sqrt{\frac{c_0 C^2 (\ln(2T/\delta) + d_2 \ln(n))}{m}}$$

We will condition on this upper-bound event on γ_A for the rest of this proof, which is the case except with probability $\delta/2$. By Theorem 5.4.9, we know that the average plays $(\overline{D}, \overline{\lambda})$ form an $(\gamma_L + \gamma_A)$-approximate equilibrium.

Finally, we need to bound the additional error for outputting the sparse approximation $\hat{D}$ instead of $\overline{D}$. We can directly apply Lemma 5.4.11, which implies that except with probability $\delta/2$, the pair $(\hat{D}, \overline{\lambda})$ form a R-approximate equilibrium, with

$$R \leq \gamma_A + \gamma_L + \frac{\sqrt{c_0 C^2 (\ln(2/\delta) + d_2 \ln(n))}}{\sqrt{m}}$$

Note that $R \leq \nu$ as long as we have $C = 1/\nu$,

$$m = \frac{(\ln(2T/\delta) d_2 \ln(n)) C^2 c_0 T}{\sqrt{n}(1 + C)^2 \ln(2/\delta)} \quad \text{and,} \quad T = \frac{4\sqrt{n} \ln(2/\delta)}{\nu^4}$$

This completes our proof. □

5.5. Experimental Evaluation

We now describe an experimental evaluation of our proposed algorithmic framework on a dataset in which fairness is a concern, due to the preponderance of racial and other sensitive features. For far more detailed experiments on four real datasets investigating the convergence properties of our algorithm, evaluating its accuracy vs. fairness tradeoffs, and comparing our approach to the recent algorithm of Agarwal et al. (2017), we direct the reader to Kearns et al. (2018). Python code and an illustrative Jupyter notebook are provided here (https://github.com/algowatchpenn/GerryFair).

While the no-regret-based algorithm described in the last section enjoys provably polynomial time convergence, for the experiments we instead implemented a simpler yet effective algorithm based on *Fictitious Play* dynamics. We first describe and discuss this modified algorithm.

5.5.1. Solving the Game with Fictitious Play

Like the algorithm given in the last section, the algorithm we implemented works by simulating a game dynamic that converges to Nash equilibrium in the zero-sum game that we derived, corresponding to the Fair ERM problem. Rather than using a no-regret dynamic, we instead use a simple iterative procedure known as *Fictitious Play* (Brown, 1949). Fictitious Play dynamics has the benefit of being more practical to implement: at each round, both players simply need to compute a single best response to the empirical play of their opponents, and this optimization requires only a single call to a CSC oracle. In contrast, the FTPL dynamic we gave in the previous section requires making many calls to a CSC oracle per round — a computationally expensive process — in order to find a sparse approximation to the Learner's mixed strategy at that round. Fictitious Play also has the benefit of being deterministic, unlike the randomized sampling required in the FTPL no-regret dynamic, thus eliminating a source of experimental variance.

The disadvantage is that Fictitious Play is only known to converge to equilibrium in the limit Robinson (1951), rather than in a polynomial number of rounds (though it is conjectured to converge quickly under rather general circumstances; see Daskalakis and Pan (2014) for a recent discussion). Nevertheless, this is the algorithm that we use in our experiments — and as we will show, it performs well on real data, despite the fact that it has weaker theoretical guarantees compared to the algorithm we presented in the last section.

Fictitious play proceeds in rounds, and in every round each player chooses a best response to his opponent's empirical history of play across previous rounds, by treating it as the

mixed strategy that randomizes uniformly over the empirical history. Pseudocode for the implemented algorithm is given below.

Algorithm 7 FairFictPlay: Fair Fictitious Play

Input: distribution $\mathcal{P}$ over the labelled data points, CSC oracles CSC($\mathcal{H}$) and CSC($\mathcal{G}$) for the classes $\mathcal{H}(S)$ and $\mathcal{G}(S)$ respectively, dual bound C, and number of rounds T
Initialize: set h^0 to be some classifier in $\mathcal{H}$, set λ^0 to be the zero vector. Let $\overline{D}$ and $\overline{\lambda}$ be the point distributions that put all their mass on h^0 and λ^0 respectively.
For $t = 1, \ldots, T$:
Compute the empirical play distributions:
[2]**Let** $\overline{D}$ be the uniform distribution over the set of classifiers $\{h^0, \ldots, h^{t-1}\}$
[2]**Let** $\overline{\lambda} = \frac{\sum_{t' \leq t} \lambda^{t'}}{t}$ be the auditor's empirical dual vector
Learner best responds: Use the oracle CSC($\mathcal{H}$) to compute $h^t = \text{argmin}_{h \in \mathcal{H}(S)} \langle \text{LC}(\overline{\lambda}), h \rangle$
Auditor best responds: Use the oracle CSC($\mathcal{G}$) to compute $\lambda^t = \text{argmax}_{\lambda} \mathbb{E}_{h \sim \overline{D}}[U(h, \lambda)]$
Output: the final empirical distribution $\overline{D}$ over classifiers

5.5.2. Description of Data

The dataset we use for our experimental valuation is known as the "Communities and Crime" (C&C) dataset, available at the UC Irvine Data Repository[4]. Each record in this dataset describes the aggregate demographic properties of a different U.S. community; the data combines socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR. The total number of records is 1994, and the number of features is 122. The variable to be predicted is the rate of violent crime in the community.

While there are larger and more recent datasets in which subgroup fairness is a potential concern, there are properties of the C&C dataset that make it particularly appealing for the initial experimental evaluation of our proposed algorithm. Foremost among these is the relatively high number of sensitive or protected attributes, and the fact that they are real-valued (since they represent aggregates in a community rather than specific individuals). This means there is a very large number of protected sub-groups that can be defined over them. There are distinct continuous features measuring the percentage or per-capita

[4] http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime

representation of multiple racial groups (including white, black, Hispanic, and Asian) in the community, each of which can vary independently of the others. Similarly, there are continuous features measuring the average per capita incomes of different racial groups in the community, as well as features measuring the percentage of each community's police force that falls in each of the racial groups. Thus restricting to features capturing race statistics and a couple of related ones (such as the percentage of residents who do not speak English well), we obtain an 18-dimensional space of real-valued protected attributes. We note that the C&C dataset has numerous other features that arguably could or should be protected as well (such as gender features), which would raise the dimensionality of the protected subgroups even further. [5]

We convert the real-valued rate of violent crime in each community to a binary label indicating whether the community is in the 70th percentile of that value, indicating that it is a relatively high-crime community. Thus the strawman baseline that always predicts 0 (lower crime) has error approximately 30% or 0.3 on this classification problem. We chose the 70th percentile since it seems most natural to predict the highest crime rates.

As in the theoretical sections of the paper, our main interest and emphasis is on the effectiveness of our proposed algorithm **FairFictPlay** on a given dataset, including:

- Whether the algorithm in fact converges, and does so in a feasible amount of computation. Recall that formal convergence is only guaranteed under the assumption of oracles that do not exist in practice, and even then is only guaranteed asymptotically.
- Whether the classifier learned by the algorithm has nontrivial accuracy, as well as strong subgroup fairness properties.
- Whether the algorithm and dataset permits nontrivial tuning of the trade-off between accuracy and subgroup fairness.

[5] Ongoing experiments on other datasets where fairness is a concern will be reported on in a forthcoming experimental paper.

As discussed in Section 5.2.1, we note that all of these issues can be investigated entirely in-sample, without concern for generalization performance. Thus for simplicity, despite the fact that our algorithm enjoys all the usual generalization properties depending on the VC dimension of the Learner's hypothesis space and the Auditor's subgroup space (see Theorems 5.2.9 and 5.2.8), we report all results here on the full C&C dataset of 1994 points, treating it as the true distribution of interest.

5.5.3. Algorithm Implementation

The main details in the implementation of **FairFictPlay** are the identification of the model classes for Learner and Auditor, the implementation of the cost sensitive classification oracle and auditing oracle, and the identification of the protected features for Auditor. For our experiments, at each round Learner chooses a linear threshold function over all 122 features. We implement the cost sensitive classification oracle via a two stage regression procedure. In particular, the inputs to the cost sensitive classification oracle are cost vectors c_0, c_1, where the i^{th} element of c_k is the cost of predicting k on datapoint i. We train two linear regression models r_0, r_1 to predict c_0 and c_1 respectively, using all 122 features. Given a new point x, we predict the cost of classifying x as 0 and 1 using our regression models: these predictions are $r_0(x)$ and $r_1(x)$ respectively. Finally we output the prediction $\hat{y}$ corresponding to lower predicted cost: $\hat{y} = \text{argmin}_{i \in \{0,1\}} r_i(x)$.

Auditor's model class consists of all linear threshold functions over just the 18 aforementioned protected race-based attributes. As per the algorithm, at each iteration t Auditor attempts to find a subgroup on which the false positive rate is substantially different than the base rate, given the Learner's randomized classifier so far. We implement the auditing oracle by treating it as a weighted regression problem in which the goal is find a linear function (which will be taken to define the subgroup) that on the negative examples, can predict the Learner's probabilistic classification on each point. We use the same regression subroutine as Learner does, except that Auditor only has access to the 18 sensitive features, rather than all 122.

Recall that in addition to the choices of protected attributes and model classes for Learner and Auditor, **FairFictPlay** has a parameter C, which is a bound on the norm of the dual variables for Auditor (the dual player). While the theory does not provide an explicit bound or guide for choosing C, it needs to be large enough to permit the dual player to force the minmax value of the game. For our experiments we chose $C = 10$, which despite being a relatively small value seems to suffice for (approximate) convergence.

The other and more meaningful parameter of the algorithm is the bound γ in the Fair ERM optimization problem implemented by the game, which controls the amount of unfairness permitted. If on a given round the subgroup disparity found by the Auditor is greater than γ, the Learner must react by adding a fairness penalty for this subgroup to its objective function; if it is smaller than γ, the Learner can ignore it and continue to optimize its previous objective function. Ideally, and as we shall see, varying γ allows us to trace out a menu of trade-offs between accuracy and fairness.

5.5.4. Results

Particularly in light of the gaps between the idealized theory and the actual implementation, the most basic questions about **FairFictPlay** are whether it converges at all, and if so, whether it converges to "interesting" models — that is, models with both nontrivial classification error (much better than the 30% or 0.3 baserate), and nontrivial subgroup fairness (much better than ignoring fairness altogether). We shall see that at least for the C&C dataset, the answers to these questions is strongly affirmative.

We begin by examining the evolution of the error and unfairness of Learner's model. In the left panel of Figure 2 we show the error of the model found by Learner vs. iteration for values of γ ranging from 0 to 0.029. Several comments are in order.

First, after an initial period in which there is a fair amount of oscillatory behavior, by 6000 iterations most of the curves have largely flattened out, and by 8,000 iterations it appears most but not all have reached approximate convergence. Second, while the top-to-bottom

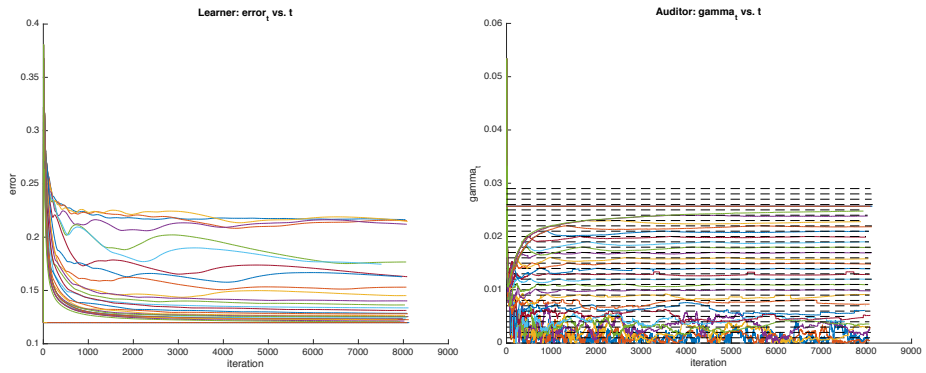


Figure 2: Evolution of the error and unfairness of Learner's classifier across iterations, for varying choices of γ. (a) Error ε_t of Learner's model vs iteration t. (b) Unfairness γ_t of subgroup found by Auditor vs. iteration t, as measured by Definition 5.2.3. See text for details.

ordering of these error curves is approximately aligned with decreasing γ — so larger γ generally results in lower error, as expected — there are many violations of this for small t, and even a few at large t. Third, and as we will examine more closely shortly, the converged values at large t do indeed exhibit a range of errors.

In the right panel of Figure 2, we show the corresponding unfairness γ_t of the subgroup found by the Auditor at each iteration t for the same runs and values of the parameter γ (indicated by horizontal dashed lines), with the same color-coding as for the left panel. Now the ordering is generally reversed — larger values of γ generally lead to higher γ_t curves, since the fairness constraint on the Learner is weaker. We again see a great deal of early oscillatory behavior, with most γ_t curves then eventually settling at or near their corresponding input γ value, as Learner and Auditor engage in a back-and-forth struggle for lower error for Learner and γ-subgroup fairness for Auditor.

For any choice of the parameter γ, and each iteration t, the two panels of Figure 2 yield a pair of realized values $\langle \varepsilon_t, \gamma_t \rangle$ from the experiment, corresponding to a Learner model whose error is ε_t, and for which the worst subgroup the Auditor was able to find had unfairness γ_t. The set of all $\langle \varepsilon_t, \gamma_t \rangle$ pairs across all runs or γ values thus represents the

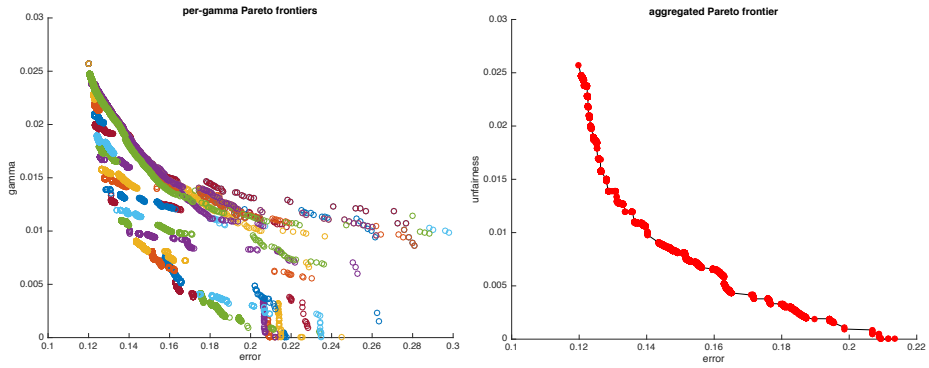


Figure 3: (a) Pareto-optimal error-unfairness values, color coded by varying values of the input parameter γ. (b) Aggregate Pareto frontier across all values of γ. Here the γ values cover the same range but are sampled more densely to get a smoother frontier. See text for details.

different trade-offs between error and unfairness found by our algorithm on the data. Most of these pairs are of course Pareto-dominated by other pairs, so we are primarily interested in the undominated frontier.

In the left panel of Figure 3, for each value of γ we show the Pareto-optimal pairs, color-coded for the value of γ. Each value of γ yields a set or cloud of undominated pairs that are usually fairly close to each other, and as expected, as γ is increased, these clouds generally move leftwards and upwards (lower error and higher unfairness).

We anticipate that the practical use of our algorithm would, as we have done, explore many values of γ and then pick a model corresponding to a point on the aggregated Pareto frontier across all γ, which represents the collection of all undominated models and the overall error-unfairness trade-off. This aggregate frontier is shown in the right panel of Figure 3, and shows a relatively smooth menu of options, ranging from error about 0.21 and no unfairness at one extreme, to error about 0.12 and unfairness 0.025 at the other, and an appealing assortment of intermediate trade-offs. Of course, in a real application the selection of a particular point on the frontier should be made in a domain-specific manner by the stakeholders or policymakers in question.

CHAPTER 6

Eliciting and Enforcing Subjective Individual Fairness

6.1. Introduction

Individual Fairness for algorithmic decision making was originally formulated as the compelling idea that "similar individuals should be treated similarly" by Dwork et al. (2012). In its original formulation, "similarity" was determined by a *task-specific metric* on individuals, which would be provided to the algorithm designer. Since then, the formulation of this task-specific fairness metric has been the primary obstacle that has stood in the way of adoption and further development of this conception of individual fairness. This is for two important reasons:

1. First, although people might have strong intuitions about what kinds of decisions are *unfair*, it is difficult for them to distill these intuitions into a concisely defined quantitative measure.

2. Second, different people disagree on what constitutes "fairness". There is no reason to suspect that even if particular individuals were able to distill their intuitive notions of fairness into some quantitative measure, that those measures would be consistent with one another, or even internally consistent.

In this chapter, we propose a practical but rigorous approach aimed at circumventing this difficulty, while staying close to the original idea that "similar individuals should be treated similarly". We are motivated by the following idea: Even if people cannot distill their conception of fairness as a quantitative metric, they can still be asked to express their opinion about whether particular pairs of individuals should be treated similarly or not. Thus, one could choose a panel of "judges", or even a particular person, and elicit opinions from them about whether certain pairs of decisions were fair or not. There is no reason to

suspect that these pairwise opinions will be consistent in any sense, or that they will form a metric. Nevertheless, once such a set of pairwise fairness constraints has been elicited, and once a data distribution and hypothesis class are fixed, there is a well-defined learning problem: minimize classification error subject to the constraint that the violation of the specified pairs is held below some fixed threshold. By varying this threshold, we can in principle define a Pareto frontier of classifiers, optimally trading off error with the elicited conception of individual fairness — without ever having to commit to a restricted class of fairness notions. We would like to find the classifiers that realize this Pareto frontier. In this paper, we solve the computational, statistical, and conceptual issues necessary to do this, and demonstrate the effectiveness of our approach via a behavioral study.

6.1.1. Results

Our Model We model individuals as having features in $\mathcal{X}$ and binary labels, drawn from some distribution $\mathcal{P}$. A committee of *judges*[1] $u \in \mathcal{U}$ has preferences that certain individuals should be treated the same way by a classifier — i.e. that the probability that they are given a positive label should be the same. We represent these preferences abstractly as a set of pairs $C_u \subseteq \mathcal{X} \times \mathcal{X}$ for each judge u, where $(x, x') \in C_u$ represents that judge u would view it as *unfair* if individuals x and x' were treated substantially differently (i.e. given a positive classification with a substantially different probability). We impose no structure on how judges form their views, or the relationship between the views of different judges — i.e. the sets $\{C_u\}_{u \in \mathcal{U}}$ are allowed to be arbitrary (for example, they need not satisfy a triangle inequality), and need not be mutually consistent. We write $C = \cup_u C_u$.

We then formulate a constrained optimization problem, that has two different "knobs" with which we can quantitatively relax our fairness constraint. Suppose that we say that a γ-fairness violation corresponds to classifying a pair of individuals $(x, x') \in C$ such that their probabilities of receiving a positive label differ by more than γ (our first knob):

[1] Though we develop our formalism as a committee of judges, note that it permits the special case of a single subjective judge, which we make use of in our behavioral study.

$|\mathbb{E}[h(x) - h(x')]| \leq \gamma$. In this expression, the expectation is taken only over the randomness of the classifier h. We might ask that for *no* pair of individuals do we have a γ-fairness violation: $\max_{(x,x') \in C} |\mathbb{E}[h(x) - h(x')]| \leq \gamma$. On the other hand, we could ask for the weaker constraint that *over a random draw of a pair of individuals*, the expected fairness violation is at most η (our second knob): $\mathbb{E}_{(x,x') \sim \mathcal{P}^2}[|h(x) - h(x')| \cdot \mathbb{1}[(x,x') \in C]] \leq \eta$. We can also combine both relaxations to ask that the in expectation over random pairs, the "excess" fairness violation, on top of an allowed budget of γ, is at most η. Subject to these constraints, we would like to find the distribution over classifiers that minimizes classification error: given a setting of the parameters γ and η, this defines a benchmark with which we would like to compete.

Our Theoretical Results Even absent fairness constraints, learning to minimize 0/1 loss (even over linear classifiers) is computationally hard in the worst case (see e.g. Feldman et al. (2012b), Feldman et al. (2009b)). Despite this, learning seems to be empirically tractable in most cases. To capture the *additional* hardness of learning subject to fairness constraints, we follow several recent papers Agarwal et al. (2017); Kearns et al. (2018) in aiming to develop *oracle efficient* learning algorithms. Oracle efficient algorithms are assumed to have access to an *oracle* (realized in experiments using a heuristic — see the next section) that can solve weighted classification problems. Given access to such an oracle, oracle efficient algorithms must run in polynomial time. We show that our fairness constrained learning problem is computationally no harder than unconstrained learning by giving such an oracle efficient algorithm (or reduction), and show moreover that its guarantees generalize from in-sample to out-of-sample in the usual way — with respect to both accuracy and the frequency and magnitude of fairness violations. Our algorithm is simple and amenable to implementation, and we use it in our experimental results.

Our Experimental Results Finally, we implement our algorithm and run a set of experiments on the COMPAS recidivism prediction dataset, using fairness constraints elicited from 43 human subjects. We establish that our algorithm converges quickly (even when

implemented with fast learning heuristics, rather than "oracles"). We also explore the Pareto curves trading off error and fairness violations for different human judges, and find empirically that there is a great deal of variability across subjects in terms of their conception of fairness, and in terms of the degree to which their expressed preferences are in conflict with accurate prediction. Finally we find that most of the difficulty in balancing accuracy with the elicited fairness constraints can be attributed to a small fraction of the reported constraints.

6.1.2. Related work

Dwork et al. (2012) first proposed the notion of individual metric-fairness that we take inspiration from, imagining fairness as a Lipschitz constraint on a randomized algorithm, with respect to some "task-specific metric" to be provided to the algorithm designer. Since the original proposal, the question of where the fairness metric should come from has been one of the primary obstacles to its adoption, and the focus of subsequent work. Zemel et al. (2013) attempt to automatically learn a representation for the data (and hence, implicitly, a similarity metric) that causes a classifier to label an equal proportion of two protected groups as positive. They provide a heuristic approach and an experimental evaluation. Kim et al. (2018) consider a group-fairness like relaxation of individual metric-fairness, asking that on average, individuals in pre-specified groups are classified with probabilities proportional to the average distance between individuals in those groups. They show how to learn such classifiers given access to an oracle which can evaluate the distance between two individuals according to the metric. Compared to our work, they assume the existence of an exact fairness metric which can be accessed using a quantitative oracle, and they use this metric to define a statistical rather than individual notion of fairness. Most related to our work, Gillen et al. (2018) assumes access to an oracle which simply identifies fairness violations across pairs of individuals. Under the assumption that the oracle is exactly consistent with a metric in a simple linear class, Gillen et al. (2018) gives a polynomial time algorithm to compete with the best fair policy in an online linear contextual bandits

problem. In contrast to the unrealistic assumptions that Gillen et al. (2018) is forced to make in order to derive a polynomial time algorithm (consistency with a simple class of metrics), we make essentially no assumptions at all on the structure of the "fairness" constraints. Ilvento (2019) studies the problem of metric learning with the goal of using only a small number of numeric valued queries, which are hard for human beings to answer, relying more on comparison queries. Finally, Rothblum and Yona (2018) prove similar generalization guarantees to ours in the context of individual-metric fairness. In the setting that they consider, the metric fairness constraint is given.

6.2. Problem formulation

Let S denote a set of labeled examples $\{z_i = (x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathcal{X}$ is a feature vector and $y_i \in \mathcal{Y}$ is a label. We will also write $S_X = \{x_i\}_{i=1}^n$ and $S_Y = \{y_i\}_{i=1}^n$. Throughout the paper, we will restrict attention to binary labels, so let $\mathcal{Y} = \{0, 1\}$. Let $\mathcal{P}$ denote the unknown distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{H}$ denote a hypothesis class containing binary classifiers $h : \mathcal{X} \to \mathcal{Y}$. We assume that $\mathcal{H}$ contains a constant classifier (which will imply that the "fairness constrained" ERM problem that we define is always feasible). We'll denote classification error of hypothesis h by $err(h, \mathcal{P}) := \Pr_{(x,y)\sim\mathcal{P}}(h(x) \neq y)$ and its empirical classification error by $err(h, S) := \frac{1}{n}\sum_{i=1}^n \mathbb{1}(h(x_i) \neq y_i)$.

We assume there is a set of one or more judges $\mathcal{U}$, such that each judge $u \in \mathcal{U}$ is identified with a set of pairs of individuals $C_u \subseteq \mathcal{X}^2$ that she thinks should be "treated similarly" i.e. ideally that for the learned classifier h, $h(x) = h(x')$ (we will ask that this hold in expectation if the classifier is randomized, and will relax it in various ways). For each pair (x, x'), let $w_{x,x'}$ be the fraction of judges who would like individual x and x' to be treated similarly – that is $w_{x,x'} = \frac{|\{u|(x,x')\in C_u\}|}{|\mathcal{U}|}$. Note that $w_{x,x'} = w_{x',x}$.

In practice, we will not have direct access to the sets of pairs C_u corresponding to the judges u, but we may ask them whether particular pairs are in this set (see Section 6.5 for details about how we actually query human subjects). We model this by imagining that we present

each judge with a random set of pairs $A \subseteq [n]^2$[2], and for each pair (x_i, x_j), ask if the pair should be treated similarly or not; we learn the set of pairs in $A \cap C_u$ for each u. Define the empirical constraint set $\hat{C}_u = \{(x_i, x_j) \in C_u\}_{\forall (i,j) \in A}$ and $\hat{w}_{x_i x_j} = \frac{|\{u | (x,x') \in \hat{C}_u\}|}{|\mathcal{U}|}$, if $(i,j) \in A$ and 0 otherwise. For simplicity, we will sometimes write w_{ij} instead of w_{x_i,x_j}. Note that $\hat{w}_{ij} = w_{ij}$ for every $(i,j) \in A$.

Our goal will be to find the distribution over classifiers from $\mathcal{H}$ that minimizes classification error, while satisfying the judges' fairness requirement C. To do so, we'll try to find D, a probability distribution over $\mathcal{H}$, that minimizes the training error and satisfy the judges' empirical fairness constraints, $\hat{C}$. For convenience, we denote D's expected classification error as $err(D, \mathcal{P}) := \mathbb{E}_{h \sim D}[err(h, \mathcal{P})]$ and likewise its expected empirical classification error as $err(D, S) := \mathbb{E}_{h \sim D}[err(h, S)]$. We say that any distribution D over classifiers satisfies (γ, η)-approximate subjective fairness if it is a feasible solution to the following constrained empirical risk minimization problem:

$$\min_{D \in \Delta\mathcal{H}, \alpha_{ij} \geq 0} err(D, S) \tag{6.1}$$

$$\text{such that } \forall (i,j) \in [n]^2 : \quad \mathbb{E}_{h \sim D}\Big[h(x_i) - h(x_j)\Big] \leq \alpha_{ij} + \gamma \tag{6.2}$$

$$\sum_{(i,j) \in [n]^2} \frac{\hat{w}_{ij} \alpha_{ij}}{|A|} \leq \eta. \tag{6.3}$$

This "Fair ERM" problem, whose feasible region we denote by $\Omega(S, \hat{w}, \gamma, \eta)$, has decision variables D and $\{\alpha_{ij}\}$, representing the distribution over classifiers and the "fairness violation" terms for each pair of training points, respectively. The parameters γ and η are constants which represent the two different "knobs" we have at our disposal to quantitatively relax the fairness constraint, in an ℓ_∞ and ℓ_1 sense respectively. To understand each of them, it helps to consider them in isolation. First, imagine that we set $\eta = 0$. γ controls

[2] We will always assume that this pair set is closed under symmetry

the worst-case disparity between the probability that any pair $(x_i, x_j) \in \hat{C}$ is classified as positive. (Note that although we have constraints for *every* pair x_i, x_j, not just those in $\hat{C}$, because $\hat{w}_{i,j} = 0$ if $(x_i, x_j) \notin \hat{C}$, a solution to the above program is free to set the slack parameter $\alpha_{i,j} = 1$ for any such pair. When $\eta = 0$, the slack parameter $\alpha_{i,j}$ is constrained to be 0 whenever $\hat{w}_{i,j} > 0$ — i.e. whenever $(x_i, x_j) \in \hat{C}$.) Next imagine that $\gamma = 0$. The parameter η controls the *expected* difference in probability that a randomly selected pair $(x_i, x_j) \in A$ is classified positively, weighted by the number of judges u who feel they should be classified the same way — i.e. the expected degree of dissatisfaction of the panel of judges $\mathcal{U}$, over the random choice of a pair of individuals and the randomness of their classification[3].

6.2.1. Fairness loss

Our goal is to develop an algorithm that will minimize its empirical error $err(D, S)$, while satisfying the empirical fairness constraints $\hat{C}$. The standard VC dimension argument states that empirical classification error will concentrate around the true classification error, and we hope to show the same kind of generalization for fairness as well. To do so, we first define fairness loss here.

For some fixed randomized hypothesis $D \in \Delta\mathcal{H}$ and w, define γ-fairness loss between a pair as

$$\Pi_{D,w,\gamma}((x, x')) = w_{x,x'} \max\left(0, \left|\mathbb{E}_{h\sim D}\left[h(x) - h(x')\right]\right| - \gamma\right)$$

For a set of pairs $M \subset \mathcal{X} \times \mathcal{X}$, the γ-fairness loss of M is defined to be:

$$\Pi_{D,w,\gamma}(M) = \frac{1}{|M|} \sum_{(x,x')\in M} \Pi_{D,w,\gamma}((x, x'))$$

This is the expected degree to which the difference in classification probability for a randomly selected pair exceeds the allowable budget γ, weighted by the fraction of judges

[3] To see this, recall that $(x_i, x_j) \in C \Rightarrow (x_j, x_i) \in C$, and so constraint 6.2 can be rewritten as $|\mathbb{E}_{h\sim D}[h(x_i) - h(x_j)]| \leq \alpha_{ij} + \gamma$, and $\hat{w}_{i,j} = 0$ if $(i, j) \notin A$, and so the sum in constraint 6.3 can equivalently be taken over A rather than $[n]^2$.

who think that the pair should be treated similarly. By construction, the empirical fairness loss is bounded by η (i.e. $\Pi_{D,w,\gamma}(M) \leq \sum_{ij} \frac{\hat{w}_{ij}\alpha_{ij}}{|A|} \leq \eta$), and we show in Section 6.4,

the empirical fairness should concentrate around the true fairness loss:

$$\Pi_{D,w,\gamma}(\mathcal{P}) := \mathbb{E}_{x,x' \sim \mathcal{P}^2}\left[\Pi_{D,w,\gamma}(x,x')\right]$$

6.2.2. Cost-sensitive classification

In our algorithm, we will make use of a cost-sensitive classification (CSC) oracle. An instance of CSC problem can be described by a set of costs $\{(x_i, c_i^0, c_i^1)\}_{i=1}^n$ and a hypothesis class, $\mathcal{H}$. c_i^0 and c_i^1 correspond to the cost of labeling x_i as 0 and 1 respectively. Invoking a CSC oracle on $\{(x_i, c_i^0, c_i^1)\}_{i=1}^n$ returns a hypothesis h^* such that

$$h^* \in \operatorname*{argmin}_{h \in \mathcal{H}} \sum_{i=1}^{n} \left(h(x_i)c_i^1 + (1 - h(x_i))\, c_i^0\right)$$

We say that an algorithm is *oracle-efficient* if it runs in polynomial time assuming access to a CSC oracle.

6.3. Empirical risk minimization

In this section, we give an oracle-efficient algorithm for approximately solving our (in-sample) constrained empirical risk minimization problem.

6.3.1. Outline of the solution

We frame the problem of solving our constrained ERM problem as finding an approximate equilibrium of a zero-sum game between a primal player and a dual player, trying to minimize and maximize respectively the Lagrangian of the constrained optimization problem.

The Lagrangian for our optimization problem is

$$\mathcal{L}(D,\alpha,\lambda,\tau) = err(D,S) + \sum_{(i,j)\in[n]^2} \lambda_{ij}\left(\mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] - \alpha_{ij} - \gamma\right) + \tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta\right) \tag{6.4}$$

For the constraint in equation (6.2), corresponding to each pair of individuals (x_i, x_j), we introduce a dual variable λ_{ij}. For the constraint (6.3), we introduce a dual variable τ. The primal player's action space is $(D,\alpha) \in (\Delta\mathcal{H}, [0,1]^{n^2})$, and the dual player's action space is $(\lambda,\tau) \in (\mathcal{R}^{n^2}, \mathcal{R})$.

Solving our constrained ERM problem equivalent to finding a minmax equilibrium of $\mathcal{L}$:

$$\operatorname*{argmin}_{(D,\alpha)\in\Omega(S,\hat{w},\gamma,\eta)} err(D,S) = \operatorname*{argmin}_{D\in\Delta\mathcal{H},\alpha\in[0,1]^{n^2}} \max_{\lambda\in\mathcal{R}^{n^2},\tau\in\mathcal{R}} \mathcal{L}(D,\alpha,\lambda,\tau)$$

Because $\mathcal{L}$ is linear in terms of its parameters, Sion's minimax theorem (Sion et al., 1958) gives us

$$\min_{D\in\Delta\mathcal{H},\alpha\in[0,1]^{n^2}} \max_{\lambda\in\mathcal{R}^{n^2},\tau\in\mathcal{R}} \mathcal{L}(D,\alpha,\lambda,\tau) = \max_{\lambda\in\mathcal{R}^{n^2},\tau\in\mathcal{R}} \min_{D\in\Delta\mathcal{H},\alpha\in[0,1]^{n^2}} \mathcal{L}(D,\alpha,\lambda,\tau).$$

By a classic result of Freund and Schapire (1996b), one can compute an approximate equilibrium by simulating "no-regret" dynamics between the primal and dual player. Our algorithm can be viewed as simulating the following *no-regret dynamics* between the primal and the dual players over T rounds. Over each of the rounds, the dual player updates dual variables $\{\lambda,\tau\}$ according to *no-regret* learning algorithms (exponentiated gradient descent (Kivinen and Warmuth, 1997) and online gradient descent (Zinkevich, 2003) respectively). At every round, the primal player then best responds with a pair $\{D,\alpha\}$ using a CSC oracle. The time-averaged play of both players converges to an approximate equilibrium of the zero-sum game, where the approximation is controlled by the regret of the dual player.

6.3.2. Primal player's best response

In each round t, given the actions chosen by the dual player $\left(\lambda^t, \tau^t\right)$, the primal player needs to best respond by choosing $\left(D^t, \alpha^t\right)$ such that

$$\left(D^t, \alpha^t\right) \in \underset{D \in \Delta\mathcal{H}, \alpha \in [0,1]^{n^2}}{\operatorname{argmin}} \mathcal{L}\left(D, \alpha, \lambda^t, \tau^t\right).$$

We do so by leveraging a CSC oracle. Given λ^t, we can set the costs as follows

$$c_i^0 = \frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(y_i \neq 0)\right] \text{ and } c_i^1 = \frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(y_i \neq 1)\right] + \left(\lambda_{ij}^t - \lambda_{ji}^t\right).$$

Then, $D^t = h^t = CSC(\{(x_i, c_i^0, c_i^1)\}_{i=1}^n)$ (we note that the best response is always a deterministic classifier h^t). As for α^t, we set $\alpha_{ij}^t = 1$ if $\tau^t \frac{w_{ij}}{|A|} - \lambda_{ij}^t \leq 0$ and 0 otherwise.

Algorithm 8 Best Response, $BEST_\rho(\lambda, \tau)$, for the primal player

Input: training examples $S = \{x_i, y_i\}_{i=1}^n$, $\lambda \in \Lambda$, $\tau \in \mathrm{T}$, CSC oracle CSC
for $i = 1, \ldots, n$ **do**
  if $y_i = 0$ **then**
    Set $c_i^0 = 0$
    Set $c_i^1 = \frac{1}{n} + \sum_{j\neq i} \lambda_{ij} - \lambda_{ji}$
  else
    Set $c_i^0 = \frac{1}{n}$
    Set $c_i^1 = \sum_{j\neq i} \lambda_{ij} - \lambda_{ji}$
$D = CSC(S, c)$
for $(i, j) \in [n]^2$ **do**

$$\alpha_{ij} = \begin{cases} 1: & \tau \frac{w_{ij}}{|A|} - \lambda_{ij} \leq 0 \\ 0: & \tau \frac{w_{ij}}{|A|} - \lambda_{ij} > 0. \end{cases}$$

Output: D, α

Lemma 6.3.1. *For fixed λ, τ, the best response optimization for the primal player is separable, i.e.*

$$\underset{D,\alpha}{\operatorname{argmin}} \mathcal{L}(D, \alpha, \lambda, \tau) = \underset{D}{\operatorname{argmin}} \mathcal{L}_{\lambda,\tau}^{\rho_1}(D) \times \underset{\alpha}{\operatorname{argmin}} \mathcal{L}_{\lambda,\tau}^{\rho_2}(\alpha),$$

where

$$\mathcal{L}^{\rho_1}_{\lambda,\tau}(D) = err(h,D) + \sum_{(i,j)\in[n]^2} \lambda_{ij}\mathbb{E}_{h\sim D}\Big[h(x_i) - h(x_j)\Big]$$

and

$$\mathcal{L}^{\rho_2}_{\lambda,\tau}(\alpha) = \sum_{(i,j)\in[n]^2} \lambda_{ij}\Big(-\alpha_{ij}\Big) + \tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij}\right)$$

Proof. First, note that α is not dependent on D and vice versa. Thus, we may separate the optimization $\operatorname{argmin}_{D,\alpha}\mathcal{L}$ as such:

$$\begin{aligned}
&\operatorname*{argmin}_{D,\alpha}\mathcal{L}(D,\alpha,\lambda,\tau)\\
&= \operatorname*{argmin}_{D,\alpha} err(D,S) + \sum_{(i,j)\in[n]^2} \lambda_{ij}\Big(\mathbb{E}_{h\sim D}[h(x_i) - h(x_j)] - \alpha_{ij} - \gamma\Big) + \tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta\right)\\
&= \operatorname*{argmin}_{D} err(D,S) + \sum_{(i,j)\in[n]^2} \lambda_{ij}\mathbb{E}_{h\sim D}\Big[h(x_i) - h(x_j)\Big] \times \sum_{(i,j)\in[n]^2} \lambda_{ij}\Big(-\alpha_{ij}\Big) + \tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij}\right)\\
&= \operatorname*{argmin}_{D} \mathcal{L}^{\rho_1}_{\lambda,\tau}(D) \times \operatorname*{argmin}_{\alpha} \mathcal{L}^{\rho_2}_{\lambda,\tau}(\alpha)
\end{aligned}$$

□

Lemma 6.3.2. *For fixed λ and τ, the output α from $BEST_\rho(\lambda,\tau)$ minimizes $\mathcal{L}^{\rho_2}_{\lambda,\tau}$*

Proof. The optimization

$$\begin{aligned}
\operatorname*{argmin}_{\alpha} \mathcal{L}^{\rho_2}_{\lambda,\tau} &= \operatorname*{argmin}_{\alpha} \sum_{(i,j)\in[n]^2} \lambda_{ij}\left(-\alpha_{ij}\right) + \tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij}\right) \\
&= \operatorname*{argmin}_{\alpha} \sum_{(i,j)\in[n]^2} -\lambda_{ij}\alpha_{ij} + \sum_{(i,j)\in[n]^2} \tau\frac{w_{ij}}{|A|}\alpha_{ij} \\
&= \operatorname*{argmin}_{\alpha} \sum_{(i,j)\in[n]^2} \alpha_{ij}\left(\tau\frac{w_{ij}}{|A|} - \lambda_{ij}\right).
\end{aligned}$$

Note that for any pair $(i,j) \in [n]^2$, the term $\alpha_{ij} \in [0,1]$. Thus, when the constant $\tau\frac{w_{ij}}{|A|} - \lambda_{ij} \leq 0$, we assign α_{ij} as the maximum bound, 1, in order to minimize $\mathcal{L}_{\rho_2}$. Otherwise, when $\tau\frac{w_{ij}}{|A|} - \lambda_{ij} > 0$, we assign α_{ij} as the minimum bound, 0. □

Lemma 6.3.3. *For fixed λ and τ, the output D from $BEST_\rho(\lambda,\tau)$ minimizes $\mathcal{L}^{\rho_1}_{\lambda,\tau}$*

Proof.

$$\begin{aligned}
\operatorname*{argmin}_{D} \mathcal{L}^{\rho_1}_{\lambda,\tau} &= \operatorname*{argmin}_{D} err(D,S) + \sum_{(i,j)\in[n]^2} \lambda_{ij}\mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] \\
&= \operatorname*{argmin}_{D} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{h\sim D}\left[\mathbb{1}(h(x_i) \neq y_i)\right] + \sum_{(i,j)\in[n]^2} \lambda_{ij}\mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] \\
&= \operatorname*{argmin}_{D} \sum_{i=1}^{n}\left(\frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(h(x_i) \neq y_i)\right] + \sum_{j\neq i}\lambda_{ij}h(x_i) - \sum_{j\neq i}\lambda_{ji}h(x_i)\right) \\
&= \operatorname*{argmin}_{D} \sum_{i=1}^{n}\left(\frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(h(x_i) \neq y_i)\right] + \sum_{j\neq i} h(x_i)\left(\lambda_{ij} - \lambda_{ji}\right)\right).
\end{aligned}$$

For each $i \in [n]$, we assign the cost

$$c_i^{h(x_i)} = \frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(h(x_i) \neq y_i)\right] + h(x_i)\left(\lambda_{ij} - \lambda_{ji}\right).$$

Note that the cost depends on whether $y_i = 0$ or 1. For example, take $y_i = 1$ and $h(x_i) = 0$. The cost

$$c_i^{h(x_i)} = c_i^0 = \frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(h(x_i) \neq y_i)\right] + \sum_{j\neq i} h(x_i)\left(\lambda_{ij} - \lambda_{ji}\right)$$
$$= \frac{1}{n}\cdot 1 + \sum_{j\neq i} 0\cdot\left(\lambda_{ij} - \lambda_{ji}\right) = \frac{1}{n}$$

□

6.3.3. Dual player's no regret updates

In order to reason about convergence we need to restrict the dual player's action space to lie within a bounded ℓ_1 ball, defined by the parameters C_τ and C_λ that appear in our theorem — and serve to trade off running time with approximation quality:

$$\Lambda = \left\{\lambda \in \mathcal{R}_+^{n^2} : \|\lambda\|_1 \leq C_\lambda\right\}, \mathrm{T} = \{\tau \in \mathcal{R}_+ : \|\tau\|_1 \leq C_\tau\}.$$

The dual player will use exponentiated gradient descent (Kivinen and Warmuth, 1997) to update λ and online gradient descent (Zinkevich, 2003) to update τ, where the reward function will be defined as: $r_\lambda\left(\lambda^t\right) = \sum_{(i,j)\in[n]^2} \lambda_{ij}^t\left(\mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] - \alpha_{ij} - \gamma\right)$ and $r_\lambda\left(\tau^t\right) = \tau^t\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta\right)$.

Lemma 6.3.4. *Running online gradient descent for τ^t, i.e.*

$$\tau^t = proj_{[0,C_\tau]}\left(\tau^{t-1} + \mu^{t-1}\cdot\nabla\mathcal{L}_{D^t,\alpha^t}^{\psi_2}\left(\tau^{t-1}\right)\right),$$

with step size $\mu^t = \frac{C_\tau}{\sqrt{T}}$ yields the following regret

$$\max_{\tau\in\mathrm{T}}\sum_{t=1}^{T}\mathcal{L}_{D^t,\alpha^t}^{\psi_2}(\tau) - \sum_{t=1}^{T}\mathcal{L}_{D^t,\alpha^t}^{\psi_2}\left(\tau^t\right) \leq C_\tau\sqrt{T}.$$

Algorithm 9 No-Regret Dynamics

Input: training examples $\{x_i, y_i\}_{i=1}^n$, bounds C_λ and C_τ, time horizon T, step sizes μ_λ and $\{\mu_\tau^t\}_T^{t=1}$,
Set $\theta_1^0 = \mathbf{0} \in \mathbb{R}^{n^2}$
Set $\tau^0 = 0$
for $t = 1, 2, \ldots, T$ **do**
  Set $\lambda_{ij}^t = C_\lambda \frac{\exp \theta_{ij}^{t-1}}{1+\sum_{i',j' \in [n]^2} \exp \theta_{i'j'}^{t-1}}$ for all pairs $(i,j) \in [n]^2$
  Set $\tau^t = proj_{[0,C_\tau]}\left(\tau^{t-1} + \mu_\tau^t \left(\frac{1}{|A|}\sum_{i,j} w_{ij}\alpha_{ij}^{t-1} - \eta\right)\right)$
  $D^t, \alpha^t \leftarrow \text{BEST}_\rho(\lambda^t, \tau^t)$
  for $(i,j) \in [n]^2$ **do**
    $\theta_{ij}^t = \theta_{ij}^{t-1} + \mu_\lambda^{t-1}\left(\mathbb{E}_{h\sim D^t}\left[h(x_i) - h(x_j)\right] - \alpha_{ij}^t - \gamma\right)$

Output: $\frac{1}{T}\sum_{t=1}^T D^t$

Proof. First, note that $\nabla \mathcal{L}_{D^t,\alpha^t}^{\psi_2}\left(\tau^{t-1}\right) = \frac{1}{W}\sum_{ij} w_{ij}\alpha_{ij}^{t-1} - \eta$ and

$$\tau^t = proj_{[0,C_\tau]}\left(\tau^{t-1} + \mu_\tau^t\left(\frac{1}{W}\sum_{ij} w_{ij}\alpha_{ij}^{t-1} - \eta\right)\right).$$

From Zinkevich (2003), we find that the regret of this online gradient descent (translated into the terms of our paper) is bounded as follows:

$$\max_{\tau \in \mathrm{T}} \sum_{t=1}^T \mathcal{L}_{D^t,\alpha^t}^{\psi_2}(\tau) - \sum_{t=1}^T \mathcal{L}_{D^t,\alpha^t}^{\psi_2}\left(\tau^t\right) \leq \frac{C_\tau^2}{2\mu_\tau^T} + \frac{\left\|\nabla \mathcal{L}_{D,\alpha}^{\psi_2}\right\|^2}{2}\sum_{t=1}^T \mu_\tau^t, \tag{6.5}$$

where the bound on our target τ term is C_τ, the gradient of our cost function at round t is $\nabla \mathcal{L}_{D^t,\alpha^t}^{\psi_2}\left(\tau^{t-1}\right)$, and the bound $\left\|\nabla \mathcal{L}_{D,\alpha}^{\psi_2}\right\| = \sup_{\tau\in\mathcal{T},\ t\in[T]}\left\|\nabla \mathcal{L}_{D^t,\alpha^t}^{\psi_2}\left(\tau^{t-1}\right)\right\|$. To prove the above lemma, we first need to show that this bound $\left\|\nabla \mathcal{L}_{D,\alpha}^{\psi_2}\right\| \leq 1$.

Since $w_{ij}, \alpha_{ij}, \eta \in [0,1]$ for all pairs (i,j), the Lagrangian $\frac{1}{|A|}\sum_{ij} w_{ij}\alpha_{ij} - \eta = \frac{\sum_{ij} w_{ij}\alpha_{ij}}{|A|} - \eta \leq 1$. For all t, the gradient

$$\left|\nabla \mathcal{L}_{D^t,\alpha^t}^{\psi_2}\left(\tau^{t-1}\right)\right| = \frac{\sum_{ij} w_{ij}\alpha_{ij}^{t-1}}{|A|} - \eta \leq 1.$$

Thus,

$$\left|\nabla \mathcal{L}_{D,\alpha}^{\psi_2}\right| \leq 1.$$

Note that if we define $\mu_\tau^t = \frac{C_\tau}{\sqrt{T}}$, then the summation of the step sizes is equal to

$$\sum_{t=1}^{T} \mu_\tau^t = C_\tau \sqrt{T}$$

Substituting these two results into inequality (6.5), we get that the regret

$$\max_{\tau \in \mathrm{T}} \sum_{t=1}^{T} \mathcal{L}_{D^t,\alpha^t}^{\psi_2}(\tau) - \sum_{t=1}^{T} \mathcal{L}_{D^t,\alpha^t}^{\psi_2}\left(\tau^t\right) \leq \frac{C_\tau^2}{2\left(C_\tau/\sqrt{T}\right)} + \frac{1}{2} C_\tau \sqrt{T} = C_\tau \sqrt{T}$$

□

Lemma 6.3.5. *Running exponentiated gradient descent for λ^t yields the following regret:*

$$\max_{\lambda \in \Lambda} \sum_{t=1}^{T} \mathcal{L}_{D^t,\alpha^t}^{\psi_1}(\lambda) - \sum_{t=1}^{T} \mathcal{L}_{D^t,\alpha^t}^{\psi_1}\left(\lambda^t\right) \leq 2C_\lambda \sqrt{T \log n}.$$

Proof. In each round, the dual player gets to charge either some (i, j) constraint or no constraint at all. In other words, he is presented with $n^2 + 1$ options. Therefore, to account for the option of not charging any constraint, we define vector $\lambda' = (\lambda, 0)$, where the last coordinate, which will always be 0, corresponds to the option of not charging any constraint.

Next, we define the reward vector ζ^t for λ'^t as

$$\zeta^t = \left(\left(\mathbb{E}_{h \sim D^t}\left[h(x_i) - h(x_j)\right] - \alpha_{ij}^t - \gamma\right)_{i,j \in [n]^2}, 0\right).$$

Hence, the reward function is

$$r(\lambda'^t) = \zeta^t \cdot \lambda'^t = \mathcal{L}_{D^t,\alpha^t}^{\psi_1}\left(\lambda^t\right).$$

The gradient of the reward function is

$$\nabla r(\lambda'^t) = \left(\left(\nabla r(\lambda^t)\right)_{i,j\in[n^2]}, 0\right) = \left(\zeta^t, 0\right)$$

Note that the ℓ_∞ norm of the gradient is bounded by 1, i.e.

$$\left\|\nabla r(\lambda'^t)\right\|_\infty \leq 1$$

because for any t, each respective component of the gradient, $\mathbb{E}_{h\sim D^t}\left[h(x_i) - h(x_j)\right] - \alpha_{ij}^t - \gamma$, is bounded by 1.

Here, by the regret bound of Kivinen and Warmuth (1997), we obtain the following regret bound:

$$\max_{\lambda\in\Lambda}\sum_{t=1}^{T}\mathcal{L}_{D^t,\alpha^t}^{\psi_1}(\lambda) - \sum_{t=1}^{T}\mathcal{L}_{D^t,\alpha^t}^{\psi_1}\left(\lambda^t\right) \leq \frac{\log n}{\mu} + \mu\left\|\lambda'\right\|_1^2\left\|\nabla r(\lambda')\right\|_\infty^2 T \leq \frac{\log n}{\mu} + \mu C_\lambda^2 T.$$

If we take $\mu = \frac{1}{C_\lambda}\sqrt{\frac{\log n}{T}}$, the regret is bounded as follows:

$$\max_{\lambda\in\Lambda}\sum_{t=1}^{T}\mathcal{L}_{D^t,\alpha^t}^{\psi_1}(\lambda) - \sum_{t=1}^{T}\mathcal{L}_{D^t,\alpha^t}^{\psi_1}\left(\lambda^t\right) \leq 2C_\lambda\sqrt{T\log n}.$$

□

6.3.4. Guarantee

Now, we appeal to Freund and Schapire (1996b) to show that our no-regret dynamics converge to an approximate minmax equilibrium of $\mathcal{L}$. Then, we show that an approximate minmax equilibrium corresponds to an approximately optimal solution to our original constrained optimization problem.

Theorem 6.3.6 (Freund and Schapire (1996b)). *Let $(D^1,\alpha^1),\ldots,(D^T,\alpha^T)$ be the primal player's sequence of actions, and $(\lambda^1,\tau^1),\ldots,(\lambda^T,\tau^T)$ be the dual player's sequence of actions. Let $\overline{D}=\frac{1}{T}\sum_{t=1}^{T}D^t$, $\overline{\alpha}=\frac{1}{T}\sum_{t=1}^{T}\alpha^t$, $\overline{\lambda}=\frac{1}{T}\sum_{t=1}^{T}\lambda^t$, and $\overline{\tau}=\frac{1}{T}\sum_{t=1}^{T}\tau^t$. Then, if the regret of the dual player satisfies*

$$\max_{\lambda\in\Lambda,\tau\in\mathrm{T}}\sum_{t=1}^{T}\mathcal{L}\left(D^t,\alpha^t,\lambda^t,\tau^t\right)-\sum_{t=1}^{T}\mathcal{L}\left(D^t,\alpha^t,\lambda^t,\tau^t\right)\leq\xi_\psi T,$$

and the primal player best responds in each round ($D^t,\alpha^t=\operatorname{argmax}_{D\in\Delta(H),\alpha\in[0,1]^{n^2}}\mathcal{L}\left(D,\alpha,\lambda^t,\tau^t\right)$), then $\left(\overline{D},\overline{\alpha},\overline{\lambda},\overline{\tau}\right)$ is an ξ_ψ-approximate solution

Remark 6.3.7. *If the primal learner's approximate best response satisfies*

$$\sum_{t=1}^{T}\mathcal{L}\left(D^t,\alpha^t,\lambda^t,\tau^t\right)-\min_{D\in\Delta(H),\alpha\in[0,1]^{n^2}}\sum_{t=1}^{T}\mathcal{L}\left(D,\alpha,\lambda^t,\tau^t\right)\leq\xi_\rho T$$

along with dual player's regret of $\xi_\rho T$, then $\left(\overline{D},\overline{\alpha},\overline{\lambda},\overline{\tau}\right)$ is an $\left(\xi_\rho+\xi_\psi\right)$-approximate solution

Theorem 6.3.8. *Let $\left(\hat{D},\hat{\alpha},\hat{\lambda},\hat{\tau}\right)$ be a v-approximate solution to the Lagrangian problem. More specifically,*

$$\mathcal{L}\left(\hat{D},\hat{\alpha},\hat{\lambda},\hat{\tau}\right)\leq\min_{D\in\Delta(\mathcal{H}),\alpha\in[0,1]^{n^2}}\mathcal{L}\left(D,\alpha,\hat{\lambda},\hat{\tau}\right)+v,$$

and

$$\mathcal{L}\left(\hat{D},\hat{\alpha},\hat{\lambda},\hat{\tau}\right)\geq\max_{\lambda\in\Lambda,\tau\in\mathrm{T}}\mathcal{L}\left(\hat{D},\hat{\alpha},\lambda,\tau\right)-v.$$

Then, $err(\hat{D},S)\leq OPT+2v$. And as for the constraints, $\mathbb{E}_{h\sim\hat{D}}\left[h(x_i)-h(x_j)\right]\leq\hat{\alpha}_{ij}+\gamma+\frac{1+2v}{C_\lambda},\forall(i,j)\in[n]^2$ and $\frac{1}{|A|}\sum_{(i,j)\in[n]^2}\hat{w}_{ij}\hat{\alpha}_{ij}\leq\eta+\frac{1+2v}{C_\tau}$.

Proof. Let $(D^*,\alpha^*)=\operatorname{argmin}_{(D,\alpha)\in\Omega(S,\hat{w},\gamma,\eta)}err(D,S)$, the optimal solution to the Fair ERM. Also, define

$$penalty_{S,w}(D,\alpha,\lambda,\tau):=\sum_{(i,j)}\lambda_{ij}\left(\mathbb{E}_{h\sim D}\left[h(x_i)-h(x_j)\right]-\alpha_{ij}-\gamma\right)+\tau\left(\frac{1}{|A|}\sum_{(i,j)}\hat{w}_{ij}\alpha_{ij}-\eta\right).$$

Note that for any D and α, $\max_{\lambda\in\Lambda,\tau\in\mathrm{T}} penalty_{S,\hat{w}}(D,\alpha,\lambda,\tau) \geq 0$ because one can always set $\lambda = 0$ and $\tau = 0$.

$$\begin{aligned}
\max_{\lambda\in\Lambda,\tau\in\mathrm{T}} \mathcal{L}\left(\hat{D},\hat{\alpha},\lambda,\tau\right) &\leq \mathcal{L}\left(\hat{D},\hat{\alpha},\hat{\lambda},\hat{\tau}\right) + v \\
&\leq \min_{D\in\Delta(\mathcal{H}),\alpha\in[0,1]^{n^2}} \mathcal{L}\left(D,\alpha\hat{\lambda},\hat{\tau}\right) + 2v \\
&\leq \mathcal{L}\left(D^*,\alpha^*,\hat{\lambda},\hat{\tau}\right) + 2v \\
&= err(D^*,S) + penalty_{S,\hat{w}}\left(D^*,\alpha^*,\hat{\lambda},\hat{\tau}\right) + 2v \\
&\leq err(D^*,S) + 2v
\end{aligned}$$

The first inequality and the third inequality are from the definition of v-approximate saddle point, and the second to last equality comes from the fact that D^*, a^* is a feasible solution.

Now, we consider two cases when$\left(\hat{D},\hat{\alpha}\right)$ is a feasible solution and when it's not.

1. $\left(\hat{D},\hat{\alpha}\right) \in \Omega(S,\hat{w},\gamma,\eta)$

   In this case, $\max_{\lambda\in\Lambda,\tau\in\mathrm{T}} penalty_{S,\hat{w}}\left(\hat{D},\hat{\alpha},\lambda,\tau\right) = 0$ because by the definition of being a feasible solution,

   $$\mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] \leq \alpha_{ij} + \gamma, \forall (i,j) \in [n]^2$$

   and

   $$\frac{1}{|A|}\sum_{(i,j)\in[n]^2} \hat{w}_{ij}\alpha_{ij} \leq \eta.$$

   Hence, $\max_{\lambda\in\Lambda,\tau\in\mathrm{T}} \mathcal{L}\left(\hat{D},\hat{\alpha},\lambda,\tau\right) = err\left(\hat{D},S\right)$. Therefore, we have $err\left(\hat{D},S\right) \leq err(D^*,S)+ 2v$.

2. $\left(\hat{D},\hat{\alpha}\right) \notin \Omega(S,\hat{w},\gamma,\eta)$

   $\max_{\lambda\in\Lambda,\tau\in\mathrm{T}} \mathcal{L}\left(\hat{D},\hat{\alpha},\lambda,\tau\right) = err\left(\hat{D},S\right) + \max_{\lambda\in\Lambda,\tau\in\mathrm{T}} penalty_{S,\hat{w}}\left(\hat{D},\hat{\alpha},\lambda,\tau\right)$. Therefore, $err\left(\hat{D},S\right) \leq err(D^*,S) + 2v$ because $\max_{\lambda\in\Lambda,\tau\in\mathrm{T}} penalty_{S,\hat{w}}\left(\hat{D},\hat{\alpha},\lambda,\tau\right) \geq 0$.

Now, we show that even when $(\hat{D}, \hat{\alpha})$ is not a feasible solution, the constraints are violated only by so much.

$$\max_{\lambda\in\Lambda,\tau\in\mathrm{T}} \mathcal{L}(\hat{D},\hat{\alpha},\lambda,\tau) = err(\hat{D},S) + \max_{\lambda\in\Lambda,\tau\in\mathrm{T}} penalty_{S,\hat{w}}(\hat{D},\hat{\alpha},\lambda,\tau) \leq err(D^*,S) + 2v$$

$$\max_{\lambda\in\Lambda,\tau\in\mathrm{T}} penalty_{S,\hat{w}}(\hat{D},\hat{\alpha},\hat{\lambda},\hat{\tau}) \leq err(D^*,S) - err(\hat{D},S) + 2v$$

$$\max_{\lambda\in\Lambda,\tau\in\mathrm{T}} penalty_{S,\hat{w}}(\hat{D},\hat{\alpha},\hat{\lambda},\hat{\tau}) \leq 1 + 2v$$

Let $\lambda^*, \tau^* = BEST_{\psi}\left(\hat{D},\hat{\alpha}\right)$, which minimizes the function as shown in Lemma A.4.2 and A.4.3 (in the appendix). Now, consider

$$\sum_{(i,j)} \lambda^*_{ij}\left(\mathbb{E}_{h\sim D}\left[h(x_i)-h(x_j)\right]-\alpha_{ij}-\gamma\right)+\tau^*\left(\frac{1}{|A|}\sum_{(i,j)}\hat{w}_{ij}\alpha_{ij}-\eta\right)\leq 1+2v$$

Say $(i^*,j^*) = \mathrm{argmax}_{(i,j)\in[n^2]}\,\mathbb{E}_{h\sim D}\left[h(x_i)-h(x_j)\right]-\alpha_{ij}-\gamma$. Note that if $\mathbb{E}_{h\sim D}\left[h(x_{i^*})-h(x_{j^*})\right]-\alpha_{i^*j^*}-\gamma>0$, then $\lambda^*_{i^*j^*}=C_\tau$ and 0 for the other coordinates and else, it's just a zero vector. Also, $\tau = C_\tau$ if $\sum_{(i,j)}\hat{w}_{ij}\alpha_{ij}-\eta>0$ and 0 otherwise. Thus,

$$\sum_{(i,j)} \lambda^*_{ij}\left(\mathbb{E}_{h\sim D}\left[h(x_i)-h(x_j)\right]-\alpha_{ij}-\gamma\right)\geq 0 \text{ and } \tau^*\left(\frac{1}{|A|}\sum_{(i,j)}\hat{w}_{ij}\alpha_{ij}-\eta\right)\geq 0$$

Therefore, we have

$$\max_{i,j\in[n]^2}\left(\mathbb{E}_{h\sim D}[h(x_i)-h(x_j)]-\alpha_{ij}-\gamma\right)\leq\frac{1+2v}{C_\lambda},$$

and

$$\frac{1}{|A|}\sum_{(i,j)\in[n]^2}\hat{w}_{ij}\hat{\alpha}_{ij}\leq\eta+\frac{1+2v}{C_\tau}$$

□

Theorem 6.3.9. *Fix parameters ν, C_τ, C_λ that serve to trade off running time with approximation error. Running Algorithm 9 for $T = \left(\frac{2C_\lambda\sqrt{\log(n)}+C_\tau}{\nu}\right)^2$ outputs a solution $(\hat{D}, \hat{\alpha})$ with the following guarantee. The objective value is approximately optimal:*

$$err(\hat{D}, S) \leq \min_{(D,\alpha)\in\Omega(S,\hat{w},\gamma,\eta)} err(D, S) + 2\nu.$$

And the constraints are approximately satisfied: $\mathbb{E}_{h\sim\hat{D}}\left[h(x_i) - h(x_j)\right] \leq \hat{\alpha}_{ij} + \gamma + \frac{1+2\nu}{C_\lambda}, \forall (i,j) \in [n]^2$ and $\frac{1}{|A|}\sum_{(i,j)\in[n]^2} \hat{w}_{ij}\hat{\alpha}_{ij} \leq \eta + \frac{1+2\nu}{C_\tau}$.

Proof. Observe that

$$\mathcal{L}(D, \alpha, \lambda, \tau) = err(D, S) + \mathcal{L}^{\psi_1}_{D,\alpha}(\lambda) + \mathcal{L}^{\psi_2}_{D,\alpha}(\tau)$$

By how we constructed $\mathcal{L}^{\psi_1}_{D,\alpha}$ and $\mathcal{L}^{\psi_2}_{D,\alpha}$, combining Lemma 6.3.4 and 6.3.5 yields

$$\begin{aligned}
&\max_{\lambda\in\Lambda,\tau\in\mathrm{T}} \sum_{t=1}^{T} \mathcal{L}\left(D^t, \alpha^t, \lambda^t, \tau^t\right) - \sum_{t=1}^{T} \mathcal{L}\left(D^t, \alpha^t, \lambda^t, \tau^t\right) \\
&= \max_{\tau\in\mathrm{T}} \sum_{t=1}^{T} \mathcal{L}^{\psi_2}_{D^t,\alpha^t}(\tau) - \sum_{t=1}^{T} \mathcal{L}^{\psi_2}_{D^t,\alpha^t}\left(\tau^t\right) + \max_{\lambda\in\Lambda} \sum_{t=1}^{T} \mathcal{L}^{\psi_1}_{D^t,\alpha^t}(\lambda) - \sum_{t=1}^{T} \mathcal{L}^{\psi_1}_{D^t,\alpha^t}\left(\lambda^t\right) \\
&\leq \xi_\psi T,
\end{aligned}$$

where $\xi_\psi = \frac{2C_\lambda\sqrt{T\log n}+C_\tau\sqrt{T}}{T}$.

Then, theorem 6.3.6 tells us that $\overline{D}, \overline{\alpha}, \overline{\lambda}, \overline{\alpha}$ form a ξ_ψ-approximate equilibrium, where $\overline{D} = \frac{1}{T}\sum_{t=1}^{T} D^t$, $\overline{\alpha} = \frac{1}{T}\sum_{t=1}^{T} \alpha^t$, $\overline{\lambda} = \frac{1}{T}\sum_{t=1}^{T} \lambda^t$, and $\overline{\tau} = \frac{1}{T}\sum_{t=1}^{T} \tau^t$. And finally, with $T =$

$\left(\frac{2C_\lambda\sqrt{\log(n)}+C_\tau}{\nu}\right)^2$ results in $\xi_\psi = \nu$, Theorem 6.3.8 gives

$$err(\hat{D}, S) \leq \min_{(D,\alpha)\in\Omega(S,\hat{w},\gamma,\eta)} err(D, S) + 2\nu.$$

And as for the constraints,

$$\mathbb{E}_{h\sim\hat{D}}\left[h(x_i) - h(x_j)\right] \leq \hat{\alpha}_{ij} + \gamma + \frac{1+2\nu}{C_\lambda}, \forall (i,j) \in [n]^2$$

and

$$\frac{1}{|A|}\sum_{(i,j)\in[n]^2} \hat{w}_{ij}\hat{\alpha}_{ij} \leq \eta + \frac{1+2\nu}{C_\tau}.$$

□

6.4. Generalization

In this section, we show that fairness loss generalizes out-of-sample. Error generalization follows from the standard VC-dimension bound, which — because it is a uniform convergece statement is unaffected by the addition of fairness constraints.

Theorem 6.4.1 (Kearns and Vazirani (1994c))**.** *Fix some hypothesis class $\mathcal{H}$ and distribution $\mathcal{P}$. Let $S \sim P^n$ be a dataset consisting of n examples $\{x_i, y_i\}_{i=1}^n$ sampled i.i.d. from $\mathcal{P}$. Then, for any $0 < \delta < 1$, with probability $1-\delta$, for every $h \in \mathcal{H}$, we have*

$$|err(h, \mathcal{P}) - err(h, S)| \leq O\left(\sqrt{\frac{VCDIM(\mathcal{H}) + log(\frac{1}{\delta})}{n}}\right)$$

Proving that the fairness loss generalizes doesn't follow immediately from a standard VC-dimension argument for several reasons: it is not linearly separable, but defined as an average over non-disjoint *pairs* of individuals in the sample. The difference between empirical fairness loss and true fairness loss of a randomized hypothesis $D \in \Delta\mathcal{H}$ is also a non-convex function of the supporting hypotheses h, and so it is not sufficient to prove

a uniform convergence bound merely for the base hypotheses in our hypothesis class $\mathcal{H}$. We circumvent these difficulties by making use of an ε-net argument, together with an application of a concentration inequality, and an application of Sauer's lemma. Briefly, we show that with respect to fairness loss, the continuous set of distributions over classifiers have an ε-net of sparse distributions. Using the two-sample trick and Sauer's lemma, we can bound the number of such sparse distributions.

6.4.1. Fairness Loss

At a high level, our argument proceeds as follows: using McDiarmid's inequality, for any *fixed* hypothesis, its empirical fairness loss concentrates around its expectation. This argument extends to an infinite family of hypotheses with bounded VC-dimension via the standard two-sample trick, together with Sauer's lemma: the only catch is that we need to use a variant of McDiarmid's inequality that applies to sampling without replacement. However, proving that the fairness loss for each fixed hypothesis h concentrates around its expectation is not sufficient to obtain the same result for arbitrary distributions over hypotheses, because the difference between a randomized classifier's fairness loss and its expectation is a non-convex function of the mixture weights. To circumvent this issue, we show that with respect to fairness loss, there is an ε-net consisting of sparse distributions over hypotheses. Once we apply Sauer's lemma and the two-sample trick, there are only finitely many such distributions, and we can union bound over them.

We begin by stating the standard version of McDiarmid's inequality:

Theorem 6.4.2 (McDiarmid's Inequality)**.** *Suppose $X_1, \ldots, X_n$ are independent and f satisfies*

$$\sup_{x_1,\ldots,x_n,\hat{x}_i} |f(x_1,\ldots,x_n) - f(x_1,\ldots,x_{i-1},\hat{x}_i,x_{i+1},\ldots,x_n)| \leq c_i.$$

Then, for any $\varepsilon > 0$,

$$\Pr_{X^1,\ldots,X^n}\left(\left|f(X_1,\ldots,X_n) - \mathbb{E}_{X_1,\ldots,X_n}[f(X_1,\ldots,X_n)]\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^n c_i^2}\right)$$

Lemma 6.4.3. *Fix a randomized hypothesis $D \in \Delta\mathcal{H}$. Over the randomness of $S \sim \mathcal{P}^n$, we have*

$$\Pr_{S\sim\mathcal{P}^n}\left(\left|\Pi_{D,w,\gamma}(S\times S) - \mathbb{E}_S\left[\Pi_{D,w,\gamma}(S\times S)\right]\right| \geq \varepsilon\right) \leq 2\exp\left(-2n\varepsilon^2\right)$$

Proof. Define a slightly modified fairness loss function that depends on each instance instead of pairs.

$$\Pi'_{D,w,\gamma}(x_1, x_2, \ldots, x_n) = \frac{1}{n^2}\sum_{(i,j)\in[n]^2} \Pi_{D,w,\gamma}\left((x_i, x_j)\right).$$

Note that $\Pi'_{D,w,\gamma}(x_1,\ldots,x_n) = \Pi_{D,w,\gamma}(S\times S)$. The sensitivity of $\Pi'_{D,w,\gamma}(x_1, x_2, \ldots, x_n)$ is $\frac{1}{n}$, so applying McDiarmid's inequality yields the above concentration. □

Now, following the argument described above, we show that the difference between empirical fairness loss over $S \times S$ and true fairness loss converges uniformly over $D \in \Delta\mathcal{H}$ with high probability.

Theorem 6.4.4. *If $n \geq \frac{2\ln(2)}{\varepsilon^2}$,*

$$\Pr_{S\sim\mathcal{P}^n}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S) - \mathbb{E}_{x,x'}\left[\Pi_{D,w,\gamma}(x,x')\right]\right| > \varepsilon\right) \leq 8\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\exp\left(\frac{-n\varepsilon^2}{32}\right)$$

where d is the VC-dimension of $\mathcal{H}$, and $k = \frac{\ln(2n^2)}{8\varepsilon^2} + 1$.

Proof. First, by linearity of expectation, we note that $\mathbb{E}_S[\Pi_{D,w,\gamma}(S\times S)] = \mathbb{E}_{x,x'}\left[\Pi_{D,w,\gamma}(x,x')\right]$. Given S, let D^*_S be some randomized classifier such that $\left|\Pi_{D^*_S,w,\gamma}(S\times S) - \mathbb{E}_{x,x'}\left[\Pi_{D^*_S,w,\gamma}(x,x')\right]\right| >$

ε; if such hypothesis does not exist, let it be some fixed hypothesis in $\mathcal{H}$.

$$
\begin{aligned}
&\Pr_{S\sim\mathcal{P}^n, S'\sim\mathcal{P}^n}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S)-\Pi_{D,w,\gamma}(S'\times S')\right|>\frac{\varepsilon}{2}\right)\\
&\geq \Pr_{S,S'}\left(\left|\Pi_{D^*_S,w,\gamma}(S\times S)-\Pi_{D^*_S,w,\gamma}(S'\times S')\right|>\frac{\varepsilon}{2}\right)\\
&\geq \Pr_{S,S'}(\left|\Pi_{D^*_S,w,\gamma}(S\times S)-\mathbb{E}_{x,x'}\left[\Pi_{D^*_S,w,\gamma}(x,x')\right]\right|>\varepsilon \text{ and}\\
&\left|\Pi_{D^*,w,\gamma}(S'\times S')-\mathbb{E}_{x,x'}\left[\Pi_{D^*,w,\gamma}(x,x')\right]\right|\leq\frac{\varepsilon}{2})\\
&=\mathbb{E}_{S,S'}[\mathbb{1}\left(\left|\Pi_{D^*_S,w,\gamma}(S\times S)-\mathbb{E}_{x,x'}\left[\Pi_{D^*_S,w,\gamma}(x,x')\right]\right|>\varepsilon\right)\cdot\\
&\mathbb{1}\left(\left|\Pi_{D^*,w,\gamma}(S'\times S')-\mathbb{E}_{x,x'}\left[\Pi_{D^*,w,\gamma}(x,x')\right]\right|\leq\frac{\varepsilon}{2}\right)]\\
&=\mathbb{E}_{S}[\mathbb{1}\left(\left|\Pi_{D^*_S,w,\gamma}(S\times S)-\mathbb{E}_{x,x'}\left[\Pi_{D^*_S,w,\gamma}(x,x')\right]\right|>\varepsilon\right)\cdot\\
&\Pr_{S'|S}\left(\left|\Pi_{D^*,w,\gamma}(S'\times S')-\mathbb{E}_{x,x'}\left[\Pi_{D^*,w,\gamma}(x,x')\right]\right|\leq\frac{\varepsilon}{2}\right)]\\
&\geq \Pr_{S}\left(\left|\Pi_{D^*_S,w,\gamma}(S\times S)-\mathbb{E}_{x,x'}\left[\Pi_{D^*_S,w,\gamma}(x,x')\right]\right|>\varepsilon)\right)\cdot\left(1-\exp\left(-\frac{n\varepsilon^2}{2}\right)\right)\\
&\geq \frac{1}{2}\Pr_{S}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S)-\mathbb{E}_{x,x'}\left[\Pi_{D,w,\gamma}(x,x')\right]\right|>\varepsilon\right)
\end{aligned}
$$

We used Lemma 6.4.3 for the second to last inequality, and the last inequality follows from the theorem's condition (i.e. $n\geq\frac{2\ln(2)}{\varepsilon^2}$) and how we defined D^*_S.

Now, imagine sampling $\overline{S}=2n$ points from $\mathcal{P}$, and uniformly choosing n points without replacement to be S and the remaining n points to be S'. This process is equivalent to sampling n points from $\mathcal{P}$ to form S and another independent set of n points from $\mathcal{P}$ to form S'.

$$\begin{aligned}
&\Pr_{\overline{S},S,S'}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S)-\Pi_{D,w,\gamma}(S'\times S')\right|>\frac{\varepsilon}{2}\right)\\
&=\sum_{\overline{S}}\Pr\left(\overline{S}\right)\Pr_{S,S'}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S)-\Pi_{D,w,\gamma}(S'\times S')\right|>\frac{\varepsilon}{2}\,\middle|\,\overline{S}\right)
\end{aligned}$$

Now, we show that the continuous set of distributions over classifiers $\Delta\mathcal{H}$ can be approximated by an ε'-net of sparse distributions. By sparse distribution, we mean uniform distributions over supports of at most k, for some fixed value k: $\hat{D}=\frac{1}{k}(h_1,\ldots,h_k)$ where $h_i\in\mathcal{H}$ for $i\in[k]$. Because by Sauer's lemma, the set of hypotheses $\mathcal{H}$ induces at most $O(n^d)$ distinct labellings of a dataset of sizse n, we need to union bound over at most $O(n^{dk})$ distinct sparse distributions.

Lemma 6.4.5. *For some fixed data sample S of size n, any $D\in\Delta\mathcal{H}$ can be approximated by some uniform mixture over $k:=\frac{2\ln(2n^2)}{\varepsilon'^2}+1$ hypotheses $\hat{D}=\frac{1}{k}\{h_1,\ldots,h_k\}$ such that for every $(x,x')\in S\times S$,*

$$\left|\mathbb{E}_{h\sim D}\left[h(x)-h(x')\right]-\mathbb{E}_{h\sim\hat{D}}\left[h(x)-h(x')\right]\right|\leq\varepsilon'.$$

Proof. Fix some $(x,x')\in S\times S$. Randomly sample k hypotheses from D: $\{h_i\}_{i=1}^k\sim D^k$. Because for each randomly drawn hypothesis $h_i\sim D$, the difference in its prediction for x and x' is exactly $\mathbb{E}_{h\sim D}[h(x)-h(x')]$, Hoeffding's inequality yields that

$$\Pr_{h_i\sim D,i\in[k]}\left(\left|\mathbb{E}_{h\sim D}\left[h(x)-h(x')\right]-\frac{1}{k}\sum_{i=1}^{k}\left[h_i(x)-h_i(x')\right]\right|>\varepsilon'\right)\leq 2\exp\left(-\frac{2k^2\varepsilon'^2}{4k}\right)=2\exp\left(-\frac{k\varepsilon'^2}{2}\right).$$

However, there are n^2 fixed pairs in $S \times S$, and if we distribute the failure property between n^2 pairs and union bound over all of them, we get

$$\Pr_{h_i \sim D, i \in [k]}\left(\max_{(x,x') \in S \times S} \left| \mathbb{E}_{h \sim D}\left[h(x) - h(x')\right] - \frac{1}{k}\sum_{i=1}^{k}\left[h_i(x) - h_i(x')\right] \right| > \varepsilon' \right) \leq 2n^2 \exp\left(-\frac{k\varepsilon'^2}{2}\right).$$

In order to achieve non-zero probability of having

$$\left| \mathbb{E}_{h \sim D}\left[h(x) - h(x')\right] - \frac{1}{k}\sum_{i=1}^{k}\left[h_i(x) - h_i(x')\right] \right| \leq \varepsilon', \forall (x,x') \in S \times S,$$

we need to make sure $2n^2 \exp\left(-\frac{k\varepsilon'^2}{2}\right) < 1$ or $k > \frac{2\ln\left(2n^2\right)}{\varepsilon'^2}$. □

Corollary 6.4.6. *For some fixed data sample S, any $D \in \Delta\mathcal{H}$ can be approximated by a uniform mixture of $k := \frac{2\ln\left(2n^2\right)}{\varepsilon'^2} + 1$ hypotheses $\hat{D} = \frac{1}{k}\{h_1, \ldots, h_k\}$ such that*

$$\left| \Pi_{D,w,\gamma}(S \times S) - \Pi_{\hat{D},w,\gamma}(S \times S) \right| \leq \varepsilon'$$

Proof. It simply follows from Lemma 6.4.5 and the fact that $\max\left(0, \mathbb{E}_{h \sim D}\left[h(x_i) - h(x_j)\right] - \gamma\right)$ is 1-Lipschitz in terms of $\mathbb{E}_{h \sim D}\left[h(x_i) - h(x_j)\right]$. □

Using Corollary 6.4.6 and Sauer's lemma that bounds the total number of possible labelings by $\mathcal{H}$ over $2n$ points by $(\frac{e \cdot 2n}{d})^d$, we have

$$\begin{aligned}
&\sum_{\overline{S}} \Pr\left(\overline{S}\right) \Pr_{S,S'}\left(\sup_{D \in \Delta\mathcal{H}} \left| \Pi_{D,w,\gamma}(S \times S) - \Pi_{D,w,\gamma}(S' \times S') \right| > \frac{\varepsilon}{2} \,\middle|\, \overline{S} \right) \\
&\leq \sum_{\overline{S}} \Pr\left(\overline{S}\right) \Pr_{S,S'}\left(\sup_{\hat{D} \in \mathcal{H}^k} \left| \Pi_{\hat{D},w,\gamma}(S \times S) - \Pi_{\hat{D},w,\gamma}(S' \times S') \right| > \frac{\varepsilon}{2} + \varepsilon' \,\middle|\, \overline{S} \right) \\
&\leq \sum_{\overline{S}} \Pr\left(\overline{S}\right) \cdot \left(\frac{e \cdot 2n}{d}\right)^{dk} \sup_{\hat{D} \in \mathcal{H}^k} \Pr_{S,S'}\left(\left| \Pi_{\hat{D},w,\gamma}(S \times S) - \Pi_{\hat{D},w,\gamma}(S' \times S') \right| > \frac{\varepsilon}{2} + \varepsilon' \,\middle|\, \overline{S} \right)
\end{aligned}$$

Now, for any $\hat{D}$, we will try to bound the probability that the difference in fairness loss between S and S' is big. We do so by union bounding over cases where both of them deviate from its mean by too much.

If $\left|\Pi_{\hat{D},w,\gamma}(S\times S) - E_{S|\overline{S}}\left[\Pi_{\hat{D},w,\gamma}(S\times S)\right]\right| \le \frac{\varepsilon}{4}+\frac{\varepsilon'}{2}$ and $\left|\Pi_{\hat{D},w,\gamma}(S'\times S') - E_{S|\overline{S}}\left[\Pi_{\hat{D},w,\gamma}(S\times S)\right]\right| \le \frac{\varepsilon}{4}+\frac{\varepsilon'}{2}$, then $\left|\Pi_{\hat{D},w,\gamma}(S\times S) - \Pi_{\hat{D},w,\gamma}(S'\times S')\right| \le \frac{\varepsilon}{2}+\varepsilon'$. In other words,

$$\begin{aligned}
&\Pr_{S,S'}\left(\left|\Pi_{\hat{D},w,\gamma}(S\times S) - \Pi_{\hat{D},w,\gamma}(S'\times S')\right| \le \frac{\varepsilon}{2}+\varepsilon' \,\middle|\, \overline{S}\right)\\
&\ge \Pr_{S,S'}\left(\left|\Pi_{\hat{D},w,\gamma}(S\times S) - E_{S|\overline{S}}\left[\Pi_{\hat{D},w,\gamma}(S\times S)\right]\right| \le \frac{\varepsilon}{4}+\frac{\varepsilon'}{2} \text{ and } \left|\Pi_{\hat{D},w,\gamma}(S'\times S') - E_{S|\overline{S}}\left[\Pi_{\hat{D},w,\gamma}(S\times S)\right]\right| \le \frac{\varepsilon}{4}+\frac{\varepsilon'}{2} \,\middle|\, \overline{S}\right.
\end{aligned}$$

Therefore, by looking at the compliment probabilities, we have

$$\begin{aligned}
&\Pr_{S,S'}\left(\left|\Pi_{\hat{D},w,\gamma}(S\times S) - \Pi_{\hat{D},w,\gamma}(S'\times S')\right| > \frac{\varepsilon}{2}+\varepsilon' \,\middle|\, \overline{S}\right)\\
&\le \Pr_{S,S'}\left(\left|\Pi_{\hat{D},w,\gamma}(S\times S) - E_{S|\overline{S}}\left[\Pi_{\hat{D},w,\gamma}(S\times S)\right]\right| > \frac{\varepsilon}{4}+\frac{\varepsilon'}{2} \text{ or } \left|\Pi_{\hat{D},w,\gamma}(S'\times S') - E_{S|\overline{S}}\left[\Pi_{\hat{D},w,\gamma}(S\times S)\right]\right| > \frac{\varepsilon}{4}+\frac{\varepsilon'}{2} \,\middle|\, \overline{S}\right)\\
&\le 2\Pr_{S}\left(\left|\Pi_{\hat{D},w,\gamma}(S\times S) - E_{S|\overline{S}}\left[\Pi_{\hat{D},w,\gamma}(S\times S)\right]\right| > \frac{\varepsilon}{4}+\frac{\varepsilon'}{2} \,\middle|\, \overline{S}\right).
\end{aligned}$$

Here, we can't appeal to McDiarmid's because S is sampled without replacement from $\overline{S}$. However, we use the stochastic covering property to show concentration for sampling without replacement (Pemantle and Peres, 2014) (a similar technique was used by Neel et al. (2018))

Definition 6.4.7 (Pemantle and Peres (2014))**.** *$Z_1,\ldots,Z_n$ satisfy the stochastic covering property, if for any $I\subset[n]$ and $a\ge a'\in\{0,1\}^I$ coordinate-wise such that $\|a'-a\|_1 = 1$, there is a coupling ν of the distributions μ,μ' of $(Z_j : j\in[n]\setminus I)$ conditioned on $Z_I = a$ or $Z_I = a'$, respectively, such that $\nu(x,y) = 0$ unless $x\le y$ coordinate-wise and $\|x-y\|_1\le 1$.*

Theorem 6.4.8 (Pemantle and Peres (2014)). *Let $(Z_1,\dots,Z_n) \in \{0,1\}$ be random variables such that $\Pr(\sum_{i=1}^n Z_i = k) = 1$ and the stochastic covering property is satisfied. Let $f : \{0,1\}^n \to \mathcal{R}$ be an c-Lipschitz function. Then, for any $\varepsilon > 0$,*

$$\Pr\left(|f(Z_1,\dots,Z_n) - \mathbb{E}[f(Z_1,\dots,Z_n)]| \geq \varepsilon\right) \leq 2\exp\left(\frac{-\varepsilon^2}{8c^2k}\right)$$

Lemma 6.4.9 (Neel et al. (2018)). *Given a set S of n points, sample $k \leq n$ elements without replacement. Let $Z_i = \{0,1\}$ indicate whether ith element has been chosen. Then, $(Z_1,\dots,Z_n)$ satisfy the stochastic covering property.*

Let $\overline{S} = \{x_1,\dots,x_{2n}\}$. If we slightly change the definition of the fairness loss so that it depends on the indicator variables $Z_1,\dots,Z_{2n}$,

$$\Pi''_{\hat{D},w,\gamma,\overline{S}}(Z_1,\dots,Z_{2n}) = \frac{1}{n^2}\sum_{i,j\in[2n]^2} Z_i Z_j \Pi_{\hat{D},w,\gamma}(x_i,x_j) = \Pi_{\hat{D},w,\gamma}(S\times S).$$

We see that $\Pi''_{\hat{D},w,\gamma,\overline{S}}$ is $\frac{1}{n}$-Lipschitz, so by theorem 6.4.8 and lemma 6.4.9, we get

$$\Pr_S\left(\left|\Pi_{\hat{D},w,\gamma}(S\times S) - E_{S|\overline{S}}\left[\Pi_{\hat{D},w,\gamma}(S\times S)\right]\right| > \frac{\varepsilon}{4} + \frac{\varepsilon'}{2} \,\middle|\, \overline{S}\right) \leq$$
$$2\exp\left(\frac{-\left(\frac{\varepsilon}{4}+\frac{\varepsilon'}{2}\right)^2}{8\frac{1}{n^2}\cdot n}\right) = 2\exp\left(\frac{-n\left(\frac{\varepsilon}{4}+\frac{\varepsilon'}{2}\right)^2}{8}\right) \quad (6.6)$$

Combining everything, we get

$$\begin{aligned}
&\Pr_S\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S)-\mathbb{E}_{x,x'}\left[\Pi_{D,w,\gamma}(x,x')\right]\right|>\varepsilon\right)\\
&\leq 2\sum_{\overline{S}}\Pr\left(\overline{S}\right)\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\sup_{\hat{D}\in\mathcal{H}^k}\Pr_{S,S'}\left(\left|\Pi_{\hat{D},w,\gamma}(S\times S)-\Pi_{\hat{D},w,\gamma}(S'\times S')\right|>\frac{\varepsilon}{2}+\varepsilon'\,\middle|\,\overline{S}\right)\\
&\leq 4\sum_{\overline{S}}\Pr\left(\overline{S}\right)\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\sup_{\hat{D}\in\mathcal{H}^k}\Pr_{S}\left(\left|\Pi_{\hat{D},w,\gamma}(S\times S)-E_{S|\overline{S}}\left[\Pi_{\hat{D},w,\gamma}(S\times S)\right]\right|>\frac{\varepsilon}{4}+\frac{\varepsilon'}{2}\,\middle|\,\overline{S}\right)\\
&\leq 8\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\exp\left(\frac{-n\left(\frac{\varepsilon}{4}+\frac{\varepsilon'}{2}\right)^2}{8}\right)
\end{aligned}$$

For convenience, we set $\varepsilon' = \frac{\varepsilon}{2}$.

□

However, in our case, instead of finding the average over all pairs in S, we calculate the fairness loss only over m randomly chosen pairs. Fixing S, if m is sufficiently large, our empirical fairness loss should concentrate around the fairness loss over all the pairs for S.

Lemma 6.4.10. *For fixed S, randomly chosen pairs $M \subset S\times S$, and randomized hypothesis D,*

$$\Pr_{M\sim(S\times S)^m}\left(\Pi_{D,w,\gamma}(M)-\Pi_{D,w,\gamma}(S\times S)\geq\varepsilon\right)\leq\exp\left(-2m\varepsilon^2\right)$$

Proof. Write a random variable $L_a = \Pi_{D,w,\gamma}((x_{2a-1},x_{2a}))$ for the fairness loss of the ath pair. Note that

$$E[L_a]=\sum_{(i,j)\in[n]^2}\frac{1}{n^2}\Pi_{D,w,\gamma}\left((x_i,x_j)\right)=\Pi_{D,w,\gamma}(S\times S),\forall a\in[|M|].$$

Therefore, by Hoeffding's inequality, we have

$$\Pr_M\left(\Pi_{D,w,\gamma}(M)-\Pi_{D,w,\gamma}(S\times S)\geq\varepsilon\right)\leq\exp\left(-2m\varepsilon^2\right).$$

□

Once again, using ε-net sparsification of $\Delta\mathcal{H}$, we show the above concentration converges uniformly over $D \in \Delta\mathcal{H}$

Lemma 6.4.11. *For fixed S and randomly chosen pairs $M \subset S \times S$,*

$$\Pr_{M\sim(S\times S)^m}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(M)-\Pi_{D,w,\gamma}(S\times S)\right|\geq\varepsilon\right)\leq\left(\frac{e\cdot 2n}{d}\right)^{dk'}\exp\left(-8m\varepsilon^2\right),$$

where $k' = \frac{2\ln(2m)}{\varepsilon^2} + 1$.

Proof.

$$\begin{aligned}
&\Pr_{M\sim(S\times S)^m}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(M)-\Pi_{D,w,\gamma}(S\times S)\right|\geq\varepsilon\right)\\
&\leq\Pr_{M\sim(S\times S)^m}\left(\sup_{\hat{D}\in\mathcal{H}^k}\left|\Pi_{\hat{D},w,\gamma}(M)-\Pi_{\hat{D},w,\gamma}(S\times S)\right|\geq\varepsilon+2\varepsilon'\right)\\
&\leq\sum_{\hat{D}\in\mathcal{H}^k}\Pr_{M\sim(S\times S)^m}\left(\left|\Pi_{\hat{D},w,\gamma}(M)-\Pi_{\hat{D},w,\gamma}(S\times S)\right|\geq\varepsilon+2\varepsilon'\right)\\
&\leq\left(\frac{e\cdot 2n}{d}\right)^{dk}\exp\left(-2m(\varepsilon+2\varepsilon')^2\right),
\end{aligned}$$

where $k = \frac{2\ln(2m)}{4\varepsilon'^2} + 1$. The last inequality is from Corollary 6.4.6 and Lemma 6.4.10. For convenience, we just set $\varepsilon' = \varepsilon/2$. □

Combining theorem 6.4.4 and lemma 6.4.11 yields the following theorem for fairness loss generalization.

Theorem 6.4.12. *Let S consists of n i.i.d points drawn from $\mathcal{P}$ and let M represent a set of m pairs randomly drawn from $S \times S$. Then we have:*

$$\begin{aligned}
&\Pr_{\substack{S\sim\mathcal{P}^n\\M\sim(S\times S)^m}}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(M)-\mathbb{E}_{(x,x')\sim\mathcal{P}^2}[\Pi_{D,w,\gamma}(x,x')]\right|>2\varepsilon\right)\\
&\leq\left(8\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\exp\left(\frac{-n\varepsilon^2}{32}\right)+\left(\frac{e\cdot 2n}{d}\right)^{dk'}\exp\left(-8m\varepsilon^2\right)\right),
\end{aligned}$$

where $k' = \frac{2\ln(2m)}{\varepsilon^2} + 1$, $k = \frac{\ln(2n^2)}{8\varepsilon^2} + 1$, and d is the VC-dimension of $\mathcal{H}$.

To interpret this theorem, note that the right hand side (the probability of a failure of generalization) begins decreasing exponentially fast in the data and fairness constraint sample parameters n and m as soon as $n \geq \Omega(d\log(n)\log(n/d))$ and $m \geq \Omega(d\log(m)\log(n/d))$.

proof of theorem 6.4.12. With probability $1 - \left(8 \cdot \left(\frac{e \cdot 2n}{d}\right)^{dk} \exp\left(\frac{-n\varepsilon^2}{32}\right) + \left(\frac{e \cdot 2n}{d}\right)^{dk'} \exp\left(-8m\varepsilon^2\right)\right)$, where $k' = \frac{2\ln(2m)}{\varepsilon^2} + 1$ and $k = \frac{\ln(2n^2)}{8\varepsilon^2} + 1$, we have

$$\sup_{D \in \Delta\mathcal{H}} \left|\Pi_{D,w,\gamma}(M) - \Pi_{D,w,\gamma}(S \times S)\right| \leq \varepsilon$$

and

$$\sup_{D \in \Delta\mathcal{H}} \left|\Pi_{D,w,\gamma}(S \times S) - \mathbb{E}_{x,x'}[\Pi_{D,w,\gamma}(x,x')\right| \leq \varepsilon.$$

Then, by triangle inequality,

$$\sup_{D \in \Delta\mathcal{H}} \left|\Pi_{D,w,\gamma}(M) - \mathbb{E}_{x,x'}[\Pi_{D,w,\gamma}(x,x')\right| \leq 2\varepsilon.$$

In other words, with probability $\left(8 \cdot \left(\frac{e \cdot 2n}{d}\right)^{dk} \exp\left(\frac{-n\varepsilon^2}{32}\right) + \left(\frac{e \cdot 2n}{d}\right)^{dk'} \exp\left(-8m\varepsilon^2\right)\right)$, we have

$$\sup_{D \in \Delta\mathcal{H}} \left|\Pi_{D,w,\gamma}(M) - \mathbb{E}_{x,x'}[\Pi_{D,w,\gamma}(x,x')\right| > 2\varepsilon.$$

□

6.5. A Behavioral Study of Subjective Fairness: Preliminary Findings

The framework and algorithm we have provided can be viewed as a potentially powerful tool for empirically studying subjective individual fairness as a *behavioral* phenomenon. In this section we describe preliminary results from a human-subject study we performed in which subjective fairness was elicited and then enforced by our algorithm.

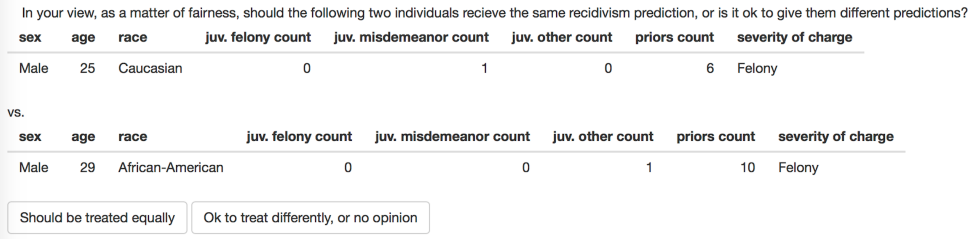


Figure 4: Screenshot of sample subjective fairness elicitation question posed to human subjects.

Our study used the COMPAS recidivism data gathered by ProPublica [4] in their celebrated analysis of Northepointe's risk assessment algorithm Angwin et al. (2016). This data consists of defendants from Broward County in Florida between 2013 to 2014. For each defendant the data consists of sex (male, female), age (18-96), race (African-American, Caucasian, Hispanic, Asian, Native American), juvenile felony count, juvenile misdemeanor count, number of other juvenile offenses, number of prior adult criminal offenses, the severity of the crime for which they were incarcerated (felony or misdemeanor), as well as the outcome of whether or not they did in fact recidivate. Recidivism is defined as a new arrest within 2 years, not counting traffic violations and municipal ordinance violations.

We implemented our fairness framework via a web app that elicited subjective fairness notions from 43 undergraduates at a major research university. After reading a document describing the data and recidivism prediction task, each subject was presented with 50 randomly chosen pairs of records from the COMPAS data set, as illustrated in Figure 4, and asked whether in their opinion the two individuals should treated (predicted) equally or not. Importantly, the subjects were shown only the features for the individuals, and not their actual recidivism outcomes, since we sought to elicit subjects' fairness notions regarding the predictions of those outcomes. While absolutely no guidance was given to subjects regarding fairness, the elicitation framework allows for rich possibilities. For example, subjects could choose to ignore demographic factors or criminal histories entirely

[4] The data can be accessed on ProPublica's Github page here. We cleaned the data as in the ProPublica study, removing any records with missing data. This left 5829 records, where the base rate of two-year recidivism was 46%.

if they liked, or a subject who believes that minorities are more vulnerable to overpolicing could discount their criminal histories relative to Caucasians in their pairwise elicitations.

For each subject, the pairs they identified to be treated equally were taken as constraints on error minimization with respect to the actual recidivism outcomes over the entire COMPAS dataset, and our algorithm was applied to solve this constrained optimization problem, using a linear threhsold heuristic as the underlying learning oracle (Kearns et al. (2018)). We ran our algorithm with $\eta = 0$ and variable γ in Equations (6.1) through (6.3), which represents the strongest enforcement of subjective fairness — the difference in predicted values must be at most γ on *every* pair selected by a subject. Because the issues we are most interested in here (convergence, tradeoffs with accuracy, and heterogeneity of fairness preferences) are orthogonal to generalization — and because we prove VC-dimension based generalization theorems — for simplicity, the results we report are in-sample.

6.5.1. *Results*

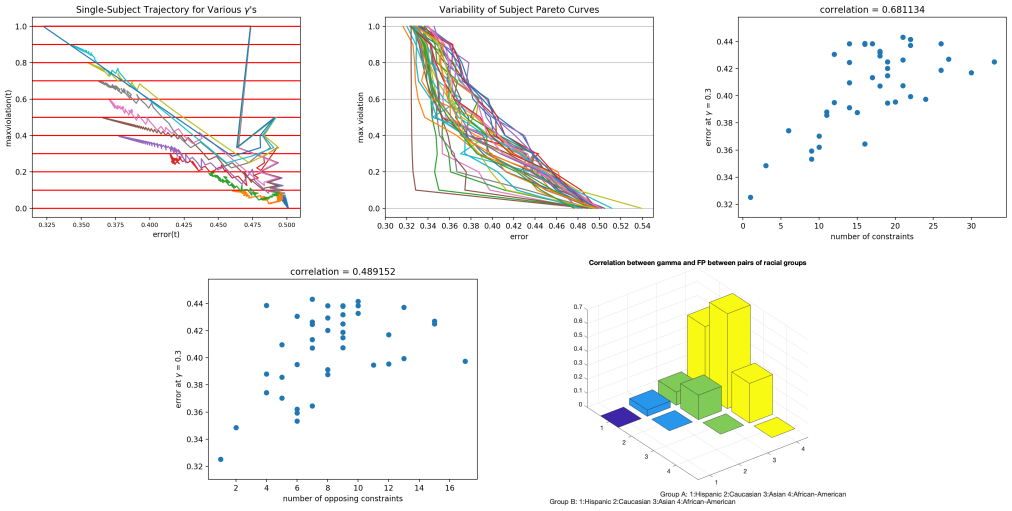


Figure 5: (a) Sample algorithm trajectory for a particular subject at various γ. (b) Sample subjective fairness Pareto curves for a sample of subjects. (c) Scatterplot of number of constraints specified and number of opposing constraints vs. error at $\gamma = 0.3$. (d) Scatterplot of number of constraints where the true labels are different vs. error at $\gamma = 0.3$. (e) Correlation between false positive rate difference and γ for racial groups.

Since our algorithm relies on a learning heuristic for which worst-case guarantees are not possible, the first empirical question is whether the algorithm converges rapidly on the behavioral data. We found that it did so consistently; a typical example is Figure 5(a), where we show the trajectory of model error vs. fairness violation for a particular subject's data in which the allowed violation was $\gamma = 0.3$ (horizontal line). At 2000 iterations the algorithm has saturated the allowed violation with the constrained error-optimal model.

Perhaps the most basic behavioral questions we might ask involve the extent and nature of subject variability. For example, do some subjects identify constraint pairs that are much harder to satifsy than other subjects? And if so, what factors seem to account for such variation?

Figure 5(b) shows that there is indeed considerable variation in subject difficulty. For a representative subset of the 43 subjects, we have plotted the error vs. fairness violation Pareto curves obtained by varying γ from 0 (pairs selected by subjects must have identical probabilistic predictions of recidivism) to 1.0 (no fairness enforced whatsoever). Since our model space is closed under probabilistic mixtures, the worst-case Pareto curve is linear, obtained by all mixtures of the error-optimal model and random predictions. Easier constaint sets are more convex. We see in the figure that both extremes are exhibited behaviorally — some subjects yield linear or near-linear curves, while others permit huge reductions in unfairness for only slight increases in error, and virtually all the possibilities in between are realized as well. [5]

Since each subject was presented with 50 random pairs and was free to constrain as many or as few as they wished, it is natural to wonder if the variation in difficulty is explained simply by the number of constraints chosen. In Figure 5(c) we show a scatterplot of the the number of constraints selected by a subject (x axis) versus the error obtained (y axis) for $\gamma = 0.3$ (an intermediate value that exhibits considerable variation in subject error rates) for all 43 subjects. While we see there is indeed strong correlation (approximately 0.69), it is

[5] The slight deviations from true convexity are due to only approximate convergence.

far from the case that the number of constraints explains all the variability. For example, amongst subjects who selected approximately 16 constraints, the resulting error varies over a range of nearly 8%, which is over 40% of the range from the optimal error (0.32) to the worst fairness-constrained error (0.5). More surprisingly, when we consider only the 'opposing' constraints, pairs of points with different true labels, the correlation (0.489) seems to be weaker. Enforcing a classifier to predict similarly on a pair of points with different true labels should increase the error, and yet, it is less correlated with error than the raw number of constraints.

It is also interesting to consider the collective force of the 1432 constraints selected by all 43 subjects together, which we can view as a "fairness panel" of sorts. Given that there are already individual subjects whose constraints yield the worst-case Pareto curve, it is unsurprising that the collective constraints do as well. But we can exploit the flexibility of our optimization framework in Equations (6.1) through constraint (6.3), and let $\gamma = 0.0$ and vary only η, thus giving the learner discretion in which subjects' constraints to discount or discard at a given budget η. In doing so we find that the unconstrained optimal error can be obtained while having the average (exact) pairwise constraint be violated by only roughly 25%, meaning roughly that only 25% of the collective constraints account for all the difficulty.

Finally, it is interesting to see if there's any relationship between subjective fairness and other standard fairness notions, such as false positive rate difference. For each subject and a pair of racial groups, we take the absolute difference in false positive rates of the classifier at $\gamma \in \{0.0, 0.1, \ldots, 1.0\}$ and calculate the correlation coefficient between γ's and the false positive rate differences. Figure 5(e) shows the average correlation coefficient across subjects for each pair of racial groups. Subjective fairness' correlation with false positive rate difference seems to be the strongest for Caucasian and African-American.

We leave the fuller investigation of our behavioral study for future work, including the detailed nature of subject variability and the comparison of behavioral subjective fairness to more standard algorithmic fairness notions.

APPENDIX

A.1. Details from Chapter 3

A.1.1. AboveThreshold

Proof of 3.2.4. Let D, D' be two neighboring databases. We will instead analyze the algorithm that outputs the entire prefix $f_1, \ldots, f_t$ when stopping at time t. Because IAT is a post-processing of this algorithm, and privacy can only be improved under post-processing, this suffices to prove the theorem. We wish to show for all outcomes $o = (t, f_1, \ldots, f_t)$:

$$\Pr[\text{IAT}(D) = (t, f_1, f_2, \ldots, f_t)] \leq e^{\varepsilon_A + \varepsilon_t} \Pr[\text{IAT}(D') = (t, f_1, f_2, \ldots, f_t)]$$

We have directly from the privacy guarantee of AboveThreshold that for every *fixed* sequence of queries $f_1, \ldots, f_t$:

$$\Pr[\text{IAT}(D) = t \mid f_1, \ldots, f_t] \leq e^{\varepsilon_A} \Pr[\text{IAT}(D') = t \mid f_1, \ldots, f_t] \tag{A.1}$$

because the guarantee of AboveThreshold is quantified over all data-independent sequences of queries $f_1, \ldots, f_T$, and by definition of the algorithm, the probability of stopping at time t is independent of the identity of any query f_t' for $t' > t$.

Now we can write:

$$\Pr[\text{IAT}(D) = t, f_1, \ldots f_t] = \Pr[\text{IAT}(D) = t \mid f_1, \ldots f_t] \Pr[M(D) = f_1, \ldots f_t]$$

By assumption, M is prefix-private, in particular, for fixed t and any $f_1, \ldots, f_t$:

$$\Pr[M(D) = f_1, \ldots f_t] \leq e^{\varepsilon_t} \Pr[M(D') = f_1, \ldots f_t]$$

Thus

$$\frac{\Pr[\text{IAT}(D) = t, f_1, \dots f_t]}{\Pr[\text{IAT}(D') = t, f_1, \dots f_t]} = \frac{\Pr[\text{IAT}(D) = t \mid f_1, \dots f_t]}{\Pr[\text{IAT}(D') = t | f_1, \dots, f_t]} \frac{\Pr[M(D) = f_1, \dots f_t]}{\Pr[M(D') = f_1, \dots f_t]}$$
$$\leq e^{\varepsilon_A} \cdot e^{\varepsilon_t} = e^{\varepsilon_A + \varepsilon_t},$$

as desired. □

We also include the following utility theorem. We say that an instantiation of AboveThreshold is (α, β) accurate with respect to a threshold W and stream of queries $f_1, \dots f_T$ if except with probability at most γ, the algorithm outputs a query f_t only if $f_t(D) \geq W - \alpha$.

Theorem A.1.1. *For any sequence of 1-sensitive queries $f_1, \dots, f_T$ such AboveThreshold is (α, β)-accurate for*

$$\alpha = \frac{8\Delta(\ln(T) + \ln(2/\gamma))}{\varepsilon}.$$

A.1.2. Doubling Method

We now describe the DoublingMethod described in Section 3.1 and Section 6.3, and give a formal ex-post privacy analysis. Let $\theta^* = \text{argmin}_{\theta \in \mathbb{R}^p} L(\theta)$. DoublingMethod accepts a list of privacy levels $\varepsilon_1 < \varepsilon_2 < \dots < \varepsilon_T$, where $\varepsilon_i = 2\varepsilon_{i-1}$. We show in A.1.12 that 2 is the optimal factor to scale ε by. It also takes in a failure probability γ, and a black-box private ERM mechanism M that has the following guarantee: Fixing a dataset D, M takes as input D and a privacy level ε_i, and generates an ε_i-differentially private hypothesis θ_i, such that the query $f^i(D) = L(D, \theta*) - L(D, \theta_i)$ has ℓ_1 sensitivity at most Δ.

Theorem A.1.2. *For $k \leq T$, define privacy loss function $\mathcal{E}(k, \theta_k) = \frac{2k\Delta \log(T/\gamma)}{\alpha} + (2^k - 1)\varepsilon_1, \varepsilon(T + 1, \theta^*) = \infty$. Then* DoublingMethod *is $\mathcal{E}$-ex-post differentially private, and is $1 - \gamma$ accurate.*

Proof. Since if the algorithm reaches step $T + 1$ it outputs the true minimizer which has error $0 < \alpha$, it could only fail to output a hypothesis with error less than α if it stops at $i \leq T$. DoublingMethod only stops early if the noisy query is greater than $-\alpha/2$; or

Algorithm 10 Doubling Method: DoublingMethod$(D, \{\varepsilon_1, \ldots, \varepsilon_T\}, M, \alpha, \gamma)$

Input: private dataset D, an accuracy α, failure probability γ, mechanism M

for each $t = 1, \ldots, T$ **do**
  Generate $\theta_t \leftarrow M(D)_t$
  Let $f^t(D) = L(D, \theta^*) - L(D, \theta_t)$
  Generate $w_t \sim \text{Lap}\left(\frac{\alpha}{2\log(\frac{T}{\gamma})}\right)$
  if $f^t(D) + w_t \geq -\alpha/2$: **then** Output (t, f^t); **Halt.**
Output $T+1, \theta^*$.

$f^i(D) + w_i \geq -\alpha/2$. But $f^i(D) \leq -\alpha$, which forces $w_i \geq \alpha/2$. By properties of the Laplace distribution, $\Pr[w_i \geq \alpha/2] = \frac{1}{2}\exp(\frac{-\alpha}{2}\frac{2\log(\frac{T}{2\gamma})}{\alpha}) = \gamma/T$. Hence by union bound over T the total failure probability is at most γ.

By the assumption, generating the k^{th} private hypothesis incurs privacy loss $\varepsilon_1 * 2^{k-1}$. By the Laplace mechanism, evaluating the error of the sensitivity Δ query f^i is $\frac{2\Delta\log(T/\gamma)}{\alpha}$-differentially private. Theorem 3.6 in Rogers et al. (2016) then says that the ex-post privacy loss of outputting $k \leq T$ is $\sum_{i=1}^{k}[\varepsilon_1 * 2^{k-1} + \frac{2\Delta\log(T/\gamma)}{\alpha}] = \frac{2k\Delta\log(T/\gamma)}{\alpha} + (2^k - 1)\varepsilon_1$, as desired. □

Remark A.1.3. *In practice, the private empirical risk minimization mechanism M may not always output a hypothesis that leads to queries with uniformly bounded ℓ_1 sensitivity. In this case a projection that scales down the hypothesis norm can be applied prior to evaluating the private query error. For a discussion of scaling the norm down refer to the experiments section of the appendix.*

A.1.3. Ridge Regression

In this subsection, we let $\ell(\theta, (X_i, y_i)) = \frac{1}{2}(y_i - \langle\theta, X_i\rangle)^2$, and the empirical loss over the data set is defined as

$$L(D, \theta) = \frac{1}{2n}\|y - X\theta\|_2^2 + \frac{\lambda\|\theta\|_2^2}{2}$$

where X denotes the $(n \times p)$ matrix with row vectors $X_1, \ldots, X_n$ and $y = (y_1, \ldots, y_n)$. We assume that for each i, $\|X_i\|_1 \leq 1$ and $|y_i| \leq 1$. For simplicity, we will sometimes write $L(\theta)$ for $L(D, \theta)$.

First, we could show that the unconstrained optimal solution in ridge regression has bounded norm.

Lemma A.1.4. *Let $\theta^* = \operatorname{argmin}_{\theta \in \mathbb{R}^d} L(\theta)$. Then $\|\theta^*\|_2 \leq \frac{1}{\sqrt{\lambda}}$.*

Proof. For any $\theta \in \mathbb{R}^p, L(\theta^*) \leq L(\theta)$. In particular for $\theta = \mathbf{0}$,

$$L(\theta^*) \leq L(\mathbf{0}) = \sum_{i=1}^{n} \frac{1}{2n} \ell((X_i, y_i), 0) \leq \frac{1}{2}$$

Note that for any $\theta, \ell((X_i, y_i), \theta) \geq 0$, so this means $L(\theta^*) \geq \frac{\lambda}{2}\|\theta^*\|_2^2$, which forces $\frac{\lambda}{2}\|\theta^*\|_2^2 \leq \frac{1}{2}$, and so $\|\theta^*\|_2 \leq \frac{1}{\sqrt{\lambda}}$ as desired. □

The following claim provides a bound on the sensitivity for the excess risk, which are the queries we send to AboveThreshold.

Claim A.1.5. *Let C be a bounded convex set in $\mathbb{R}^p$ with $\|C\|_2 \leq M$. Let D and D' be a pair of adjacent datasets, and let $\theta^* = \operatorname{argmin}_{\theta \in C} L(\theta, D)$ and $\theta^\bullet = \operatorname{argmin}_{\theta \in C} L(\theta, D')$. Then for any $\theta \in C$,*

$$|(L(\theta, D) - L(\theta^*, D)) - (L(\theta, D') - L(\theta^\bullet, D'))| \leq \frac{(M+1)^2}{n}.$$

The following lemma provides a bound on the ℓ_1 sensitivity for the matrix $X^\intercal X$ and vector $X^\intercal y$.

Lemma A.1.6. *Fix any $i \in [n]$. Let X and Z be two $n \times p$ matrices such that for all rows $j \neq i$, $X_j = Z_j$. Let $y, y' \in \mathbb{R}^n$ such that $y_j = y'_j$ for all $j \neq i$. Then*

$$\|X^\intercal X - Z^\intercal Z\|_1 \leq 2 \quad \text{and} \quad \|X^\intercal y - Z^\intercal y'\|_1 \leq 2,$$

as long as $\|X_i\|, \|Z_i\|, |y_i|, |y'_i| \leq 1$.

Proof. We can write

$$\begin{aligned}\|X^\intercal X - Z^\intercal Z\|_1 &= \|\sum_j \left(X_j^\intercal X_j - Z_j^\intercal Z_j\right)\|_1 \\ &= \|X_i^\intercal X_i - Z_i^\intercal Z\|_1 \\ &\leq \|X_i^\intercal X_i\|_1 + \|Z_i^\intercal Z_i\|_1 \\ &= \|X_i\|_1^2 + \|Z_i\|_1^2 \leq 2.\end{aligned}$$

Similarly,

$$\begin{aligned}\|X^\intercal y - Z^\intercal y'\|_1 &= \|\sum_j \left(y_j X_j - y_j' Z_j\right)\|_1 \\ &= \|y_i X_i - y_i' Z_i\|_1 \\ &= \|y_i X_i\|_1 + \|y_i' Z_i\|_1 \\ &= \|X_i\|_1 + \|Z_i\|_1 \leq 2.\end{aligned}$$

This completes the proof. □

Before we proceed to give a formal proof for 3.3.1, we will also give the following basic fact about Laplace random vectors.

Claim A.1.7. *Let $v = (v_1, \ldots, v_k) \in \mathbb{R}^k$ such that each v_i is an independent random variable drawn from the Laplace distribution $Lap(r)$. Then $\mathbb{E}[\|v\|_2] \leq \sqrt{2k}r$.*

Proof. By Jensen's inequality,

$$\mathbb{E}[\|v\|_2] = \mathbb{E}\left[\sqrt{\sum_i v_i^2}\right] \leq \sqrt{\mathbb{E}\left[\sum_i v_i^2\right]}$$

Note that by linearity of expectation and the variance of the Laplace distribution

$$\mathbb{E}\left[\sum_i \nu_i^2\right] = \sum_i \mathbb{E}\left[\nu_i^2\right] = \sum_i 2r^2 = 2kr^2.$$

Therefore, we have $\mathbb{E}\left[\|\nu\|_2\right] \le \sqrt{2k}r$. □

Proof of 3.3.1. In the algorithm, we compute $Z = X^\intercal X + B$ and $z = X^\intercal y + b$, where the entries of B and b are drawn i.i.d. from $\mathrm{Lap}(4/\varepsilon)$. Note that the output θ_p is simply a post-processing of the noisy matrix Z and vector z. Furthermore, by A.1.6, the joint vector (Z, z) is has a sensitivity bounded by 4 with respect to ℓ_1 norm. Therefore, the mechanism satisfies ε-differential privacy by the privacy guarantee of the Laplace mechanism.

Let $M = \sqrt{1/\lambda}$ and $L_p(\theta) = \frac{1}{2n}(-2\langle z, \theta\rangle) + \frac{1}{2n}(\theta^\intercal Z\theta) + \frac{\lambda\|\theta\|_2^2}{2}$. Observe that $\theta_p = \operatorname{argmin}_{\theta\in C} L_p(\theta)$. Our goal is to bound $L(\theta_p) - L(\theta^*)$, which can be written as follows

$$\begin{aligned}
L(\theta_p) - L(\theta^*) &= L(\theta_p) - L_p(\theta_p) + L_p(\theta_p) - L_p(\theta^*) + L_p(\theta^*) - L(\theta^*) \\
&\le L(\theta_p) - L_p(\theta_p) + L_p(\theta^*) - L(\theta^*) \\
&= \frac{1}{2n}\left(2\langle b, \theta_p\rangle - \theta_p^\intercal B\theta_p\right) - \frac{1}{2n}\left(2\langle b, \theta^*\rangle - (\theta^*)^\intercal B\theta^*\right)
\end{aligned}$$

Moreover, $\langle b, \theta_p\rangle \le \|b\|_2\|\theta_p\|_2 \le M\|b\|_2$ and

$$\begin{aligned}
-\theta_p^\intercal B\theta_p &= -\sum_{(s,t)\in[p]^2} B_{st}(\theta_p)_s(\theta_p)_t \\
&\le \left(\sum_{(s,t)} B_{st}^2\right)^{1/2}\left(\sum_{s,t}(\theta_p)_s^2(\theta_p)_t^2\right)^{1/2} \\
&= \|B\|_F\left[\left(\sum_s (\theta_p)_s^2\right)^2\right]^{1/2} \le \|B\|_F M^2
\end{aligned}$$

By A.1.7, we also have $\mathbb{E}[\|B\|_F] \le 4\sqrt{2}p/\varepsilon$ and $\mathbb{E}[\|b\|_2] \le 4\sqrt{2p}/\varepsilon$. Finally,

$$\begin{aligned}\mathbb{E}\Big[L(\theta_p) - L(\theta^*)\Big] &\le \mathbb{E}\left[\frac{1}{2n}\Big(2\langle b, \theta_p\rangle - \theta_p^\intercal B\theta_p\Big) - \frac{1}{2n}\left(2\langle b, \theta^*\rangle - (\theta^*)^\intercal B\theta^*\right)\right] \\ &= \mathbb{E}\left[\frac{2\langle b, \theta_p\rangle - \theta_p^\intercal B\theta_p}{2n}\right] \\ &\le \frac{\mathbb{E}[2M\|b\|_2] + \mathbb{E}\Big[M^2\|B\|_F\Big]}{2n} \le \frac{4\sqrt{2}(2\sqrt{p}M + pM^2)}{n\varepsilon}\end{aligned}$$

which recovers our stated bound. □

A.1.4. Logistic Regression

In this subsection, the input data D consists of n labelled examples $(X_1, y_1), \ldots, (X_n, y_n)$, such that for each i, $x_i \in \mathbb{R}^p$, $\|x_i\|_1 \le 1$, and $y_i \in \{-1, 1\}$.

We consider the logistic loss function: $\ell(\theta, (X_i, y_i)) = \log(1+\exp(-y_i\theta^\intercal X_i))$, and our empirical loss is defined as

$$L(\theta, D) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i\theta^\intercal X_i)) + \frac{\lambda\|\theta\|_2^2}{2}.$$

In output perturbation, the noise needs to scale with the ℓ_1-sensitivity of the optimal solution, which is given by the following lemma.

Lemma A.1.8. *Let D and D' be a pair of neighboring datasets. Let $\theta = \operatorname{argmin}_{w\in\mathbb{R}^p} L(w, D)$ and $\theta' = \operatorname{argmin}_{w'\in\mathbb{R}^p} L(w', D')$. Then $\|\theta - \theta'\|_1 \le \frac{2\sqrt{p}}{n\lambda}$.*

Proof of A.1.8. By the Corollary 8 of Chaudhuri et al. (2011), we can bound

$$\|\theta - \theta'\|_2 \le \frac{2}{n\lambda}$$

By the fact that $\|a\|_1 \le \sqrt{p}\|a\|_2$ for any $a \in \mathbb{R}^p$, we recover the stated result. □

We will show that the optimal solution for the unconstrained problem has ℓ_2 norm no more than $\sqrt{2\log 2/\lambda}$.

Claim A.1.9. *The (unconstrained) optimal solution θ^* has norm $\|\theta^*\|_2 \leq \sqrt{\frac{2\log 2}{\lambda}}$.*

Proof. Note that the weight vector $\theta = \vec{0}$ has loss $\log 2$. Therefore, $L(\theta^*) \leq \log 2$. Since the logistic loss is positive, we know that the regularization term

$$\frac{\lambda}{2}\|\theta^*\|_2^2 \leq \log 2.$$

It follows that $\|\theta^*\|_2 \leq \sqrt{\frac{2\log 2}{\lambda}}$. □

We will focus on generating hypotheses θ within the set $C = \{a \in \mathbb{R}^p \mid \|a\|_2 \leq \sqrt{2\log 2/\lambda}\}$. Then we can bound the ℓ_1 sensitivity of the excess risk using the following result.

Claim A.1.10. *Let D and D' be a pair of neighboring datasets. Then for any $\theta \in \mathbb{R}^p$ such that $\|\theta\|_2 \leq M$,*

$$|L(\theta, D) - L(\theta, D')| \leq \frac{2}{n}\log\left(\frac{1+\exp(M)}{1+\exp(-M)}\right)$$

The following fact is useful for our utility analysis for the output perturbation method.

Claim A.1.11. *Fix any data point (x, y) such that $\|x\|_1 \leq 1$ and $y \in \{-1, 1\}$. The logistic loss function $\ell(\theta, (x, y))$ is a 1-Lipschitz function in θ.*

Proof of 3.3.3. The privacy guarantee follows directly from the use of Laplace mechanism and the ℓ_1-sensitivity bound in A.1.8. Since the logistic loss function is 1-Lipschitz. For any (x, y) in our domain,

$$|\ell(\theta^*, (x, y)) - \ell(\theta^p, (x, y))| \leq \|\theta^* - \theta^p\|_2 = \|b\|_2.$$

Furthermore,

$$\|\theta_p\|_2^2 - \|\theta^*\|_2^2 = \|\theta^* + b\|_2^2 - \|\theta^*\|_2^2 = 2\langle b, \theta^*\rangle + \|b\|_2^2$$

By A.1.7 and the property of the Laplace distribution, we know that

$$\mathbb{E}\left[\|b\|_2\right] \leq \sqrt{2p}r \qquad \text{and} \qquad \mathbb{E}\left[\|b\|_2^2\right] = 2pr^2.$$

It follows that

$$\begin{aligned}\mathop{\mathbb{E}}_{b}\left[L(\theta_p) - L(\theta^*)\right] &\leq \mathop{\mathbb{E}}_{b}\left[\|b\|_2\right] + \frac{\lambda}{2}\mathbb{E}\left[\|b\|_2^2\right] \\ &\leq \sqrt{2p}r + p\lambda r^2 = \frac{2\sqrt{2}pr}{n\lambda\varepsilon} + \frac{4p^2}{n^2\lambda\varepsilon^2},\end{aligned}$$

which recovers the stated bound. □

We include the full details of OutputNR in 11.

Algorithm 11 Output Perturbation with Noise-Reduction: $\textsc{OutputNR}(D, \{\varepsilon_1, \ldots, \varepsilon_T\}, \alpha, \gamma)$

Input: private data set $D = (X, y)$, accuracy parameter α, privacy levels $\varepsilon_1 < \varepsilon_2 < \ldots < \varepsilon_T$, and failure probability γ
Let $M = \sqrt{2\log 2/\lambda}$
Instantiate Interactive AboveThreshold:
$\mathcal{A} = (D, \varepsilon_0, \alpha/2, 2\log(1+\exp(M))/(1+\exp(-M))/(n), \cdot)$
with $\varepsilon_0 = 16\Delta(\log(2T/\gamma))/\alpha$ and $\Delta = 2\log(1+\exp(M))/(1+\exp(-M))/(n)$
Let $C = \{a \in \mathbb{R}^p \mid \|a\|_2 \leq \sqrt{1/\lambda}\}$ and $\theta^* = \operatorname{argmin}_{\theta\in\mathbb{R}^p} L(\theta)$
Generate hypotheses: $\{\theta^t\} = \mathrm{NR}(\theta^*, \frac{2\sqrt{p}}{n\lambda}, \{\varepsilon_1, \ldots, \varepsilon_T\})$
for $t = 1, \ldots, T$: **do**
  if $\|\theta^t\|_2 \leq M$ **then** Set $\theta^t = M(\theta^t/\|\theta^t\|_2)$ ▷ Rescale the norm for bounded sensitivity
  Let $f^t(D) = L(D, \theta^*) - L(D, \theta^t)$
  Query $\mathcal{A}$ with f^t
  if yes **then Output** (t, θ^t)
Output: $(\perp, \theta^*)$

A.1.5. Experiments

Parameters and data

For simplicity and to avoid over-fitting, we fixed the following parameters for both experiments:

- n =100,000 (number of data points)
- $\lambda = 0.005$ (regularization parameter)
- $\gamma = 0.10$ (requested failure probability)
- $\varepsilon_1 = 4E$, where E is the inversion of the theory guarantee for the underlying algorithm. For example in the logistic regression setting where the algorithm is Output Perturbation, E is the value such that setting $\varepsilon = E$ guarantees **expected** excess risk of at most α.
- $\varepsilon_T = 1.0/n$.
- $\alpha = 0.005, 0.010, 0.015, \ldots, 0.200$ (requested excess error bound).

For NoiseReduction, we choose $T = 1000$ (maximum number of iterations) and set $\varepsilon_t = \varepsilon_1 r^t$ for the appropriate r, i.e. $r = \left(\frac{\varepsilon_T}{\varepsilon_1}\right)^{1/T}$.

For the Doubling method, T is equal to the number of doubling steps until ε_t exceeds ε_T, i.e. $T = \lceil \log_2(\varepsilon_1/\varepsilon_T) \rceil$.

Features, labels, and transformations. The Twitter dataset has $p = 77$ features (dimension of each x), relating to measurements of activity relating to a posting; the label y is a measurement of the "buzz" or success of the posting. Because general experience suggests that such numbers likely follow a heavy-tailed distribution, we transformed the labels by $y \mapsto \log(1 + y)$ and set the taks of predicting the transformed label.

The KDD-99 Cup dataset has $p = 38$ features relating to attributes of a network connection such as duration of connection, number of bytes sent in each direction, binary attributes, etc. The goal is to classify connections as innocent or malicious, with malicious connections broken down into further subcategories. We transformed three attributes containing likely heavy-tailed data (the first three mentioned above) by $x_i \mapsto \log(1 + x_i)$, dropped three

columns containing textual categorical data, and transformed the labels into 1 for any kind of malicious connection and 0 for an innocent one. (The feature length $p = 38$ is after dropping the text columns.)

For both datasets, we transformed the data by renormalizing to maximum $L1$-norm 1. That is, we computed $M = \max_i \|x_i\|_1$, and transformed each $x_i \mapsto x_i/M$. In the case of the Twitter dataset, we did the same (separately) for the y labels. This is *not* a private operation (unlike the previous ones) on the data, as it depends precisely on the maximum norm. We do not consider the problem of privately ensuring bounded-norm data, as it is orthogonal to the questions we study.

The code for the experiments is implemented in python3 using the numpy and scikit-learn libraries.

Additional results

Figure 6 plots the empirical accuracies of the output hypotheses, to ensure that the algorithms are achieving their theoretical guarantees. In fact, they do significantly better, which is reasonable considering the private testing methodology: set a threshold significantly below the goal α, add independent noise to each query, and accept only if the query plus noise is smaller than the threshold. Combined with the requirement to use tail bounds, the accuracies tend to be significantly smaller than α and with significantly higher probability than $1 - \gamma$. (Recall: this is not necessarily a good thing, as it probably costs a significant amount of extra privacy.)

Figure 7 shows the breakdown in privacy losses between the "privacy test" and the "hypothesis generator". In the case of NoiseReduction, these are AboveThreshold's ε_A and the ε_t of the private method, Covariance Perturbation or Output Perturbation. In the case of Doubling, these are the accrued ε due to tests at each step and due to Covariance Perturbation or Output Perturbation for outputting the hypotheses.

This shows the majority of the privacy loss is due to testing for privacy levels. One reason why might be that the cost of privacy tests depends heavily on certain constants, such as the norm of the hypothesis being tested. This norm is upper-bounded by a theoretical maximum which is used, but a smaller maximum would allow for significantly higher computed privacy levels for the same algorithm. In other words, the analysis might be loose compared to an analysis that knows the norms of the hypotheses, although this is a private quantity. Figure 8 supports the conclusion that generally, the theoretical maximum was very pessimistic in our cases. Note that a tenfold reduction in norm gives a tenfold reduction in privacy level for logistic regression, where sensitivity is linear in maximum norm; and a *hundred-fold* reduction for ridge regression.

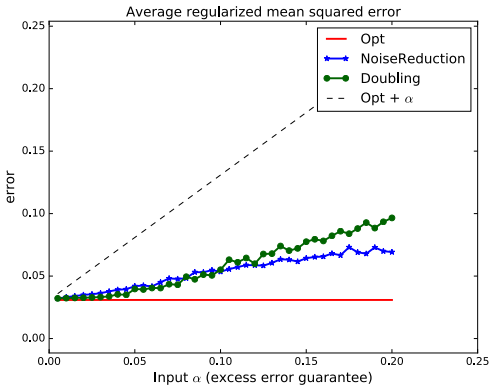
Supporting theory

Claim A.1.12. *For the "doubling method", the factor 2 increase in ε at each time step gives the optimal worst case ex post privacy loss guarantee.*

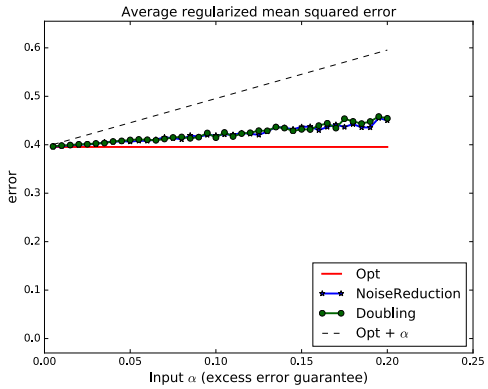
Proof. In a given setting, suppose ε^* is the "final" level of privacy at which the algorithm would halt. With a factor $1/r$ increase for $r < 1$, the final loss may be as large as ε^*/r. The total loss is the sum of that loss and all previous losses, i.e. if t steps were taken:

$$\begin{aligned}(\varepsilon^*/r) + r \cdot (\varepsilon^*/r) + \cdots + r^{t-1} \cdot (\varepsilon^*/r) &= (\varepsilon^*/r) \sum_{j=0}^{t-1} r^j \\ &\to (\varepsilon^*/r) \sum_{j=0}^{\infty} r^j \\ &= \frac{\varepsilon^*}{r(1-r)} \\ &\geq 4\varepsilon^*.\end{aligned}$$

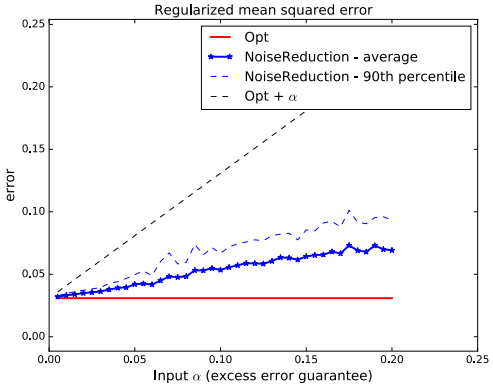
The final inequality implies that setting $r = 0.5$ and $(1/r) = 2$ is optimal. The asymptotic $\to$ is justified by noting that the starting ε_1 may be chosen arbitrarily small, so there exist



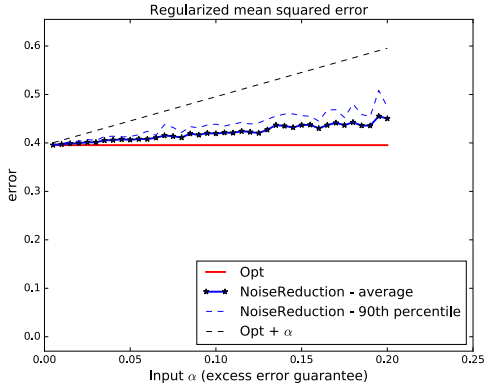
(a) **Linear (ridge) regression.**



(b) **Regularized logistic regression.**

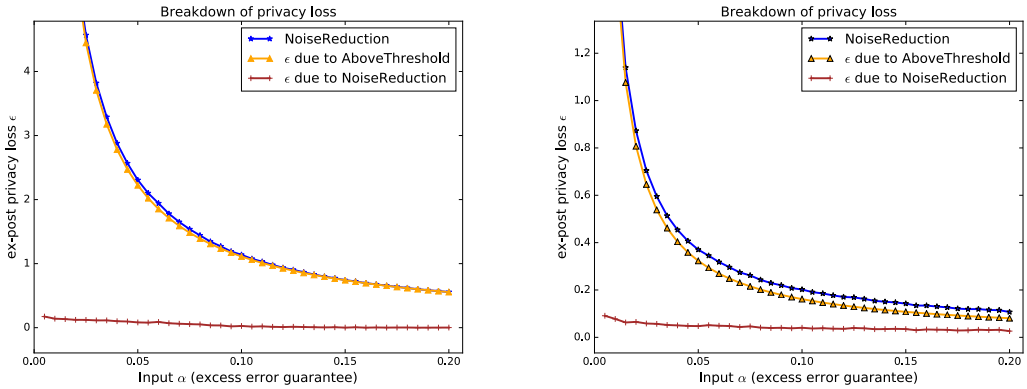


(c) **Linear (ridge) regression.**



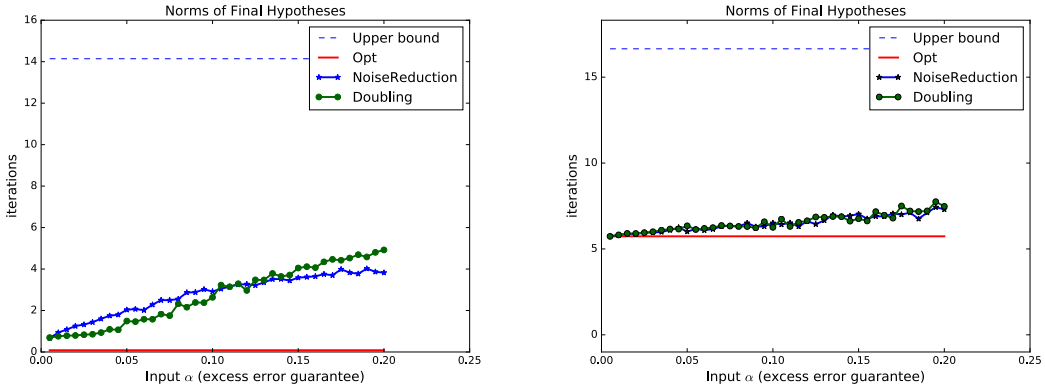
(d) **Regularized logistic regression.**

Figure 6: **Empirical accuracies.** The dashed line shows the requested accuracy level, while the others plot the actual accuracy achieved. Due most likely due to a pessimistic analysis and the need to set a small testing threshold, accuracies are significantly better than requested for both methods.



(a) **Linear (ridge) regression.** (b) **Regularized logistic regression.**

Figure 7: **Privacy breakdowns.** Shows the amount of empirical privacy loss due to the AboveThreshold versus the losses due to computing the hypotheses.



(a) **Linear (ridge) regression.** (b) **Regularized logistic regression.**

Figure 8: L_2 **norms of final hypotheses.** Shows the average L_2 norm of the output $\hat{\theta}$ for each method, versus the theoretical maximum of $1/\sqrt{\lambda}$ in the case of ridge regression and $\sqrt{2\log(2)/\lambda}$ in the case of regularized logistic regression.

parameters that exceed the value of that summation for any finite t; and the summation limits to $\frac{1}{1-r}$ as $t \to \infty$. □

A.2. Details from Chapter 4

A.2.1. Examples of Separator Sets

Separator Sets for Empirical Loss Queries.

In this section we show how to construct a separator set for a family of empirical loss queries defined over a hypothesis class, given a separator set for the corresponding hypothesis class. Formally, let us consider the data domain $\mathcal{X}$ to be the set of labelled examples $\mathcal{X}_A \times \{0,1\}$, where $\mathcal{X}_A$ is the domain of *attribute vectors*. Let $\mathcal{H}$ be a hypothesis class, with each hypothesis $h : \mathcal{X}_a \to \{0,1\}$ mapping attribute vectors to binary labels. Let $U_{\mathcal{H}}$ be a separator set for $\mathcal{H}$ such that for any pair of distinct hypotheses $h, h' \in \mathcal{H}$, there is a $u \in U_{\mathcal{H}}$ such that $h(u) \neq h'(u)$.

For every $h \in \mathcal{H}$, let $q_h \colon \mathcal{X} \to \{0,1\}$ be the *loss query* corresponding to h such that $q_h((x,y)) = \mathbb{1}[h(x) \neq y]$. Define the query class $\mathcal{Q}_{\mathcal{H}}$ to be $\{q_h \mid h \in \mathcal{H}\}$, and let $U = \{(u,0) \mid u \in U_{\mathcal{H}}\}$. Solving the learning problem over $\mathcal{H}$ corresponds to solving the minimization problem over $\mathcal{Q}_{\mathcal{H}}$.

Claim A.2.1. *The set U is a separator set for the query class $\mathcal{Q}_{\mathcal{H}}$.*

Proof. Let $q_h, q_{h'} \in \mathcal{Q}_{\mathcal{H}}$ be a pair of distinct queries. Since $U_{\mathcal{H}}$ is a separator set for $\mathcal{H}$, there exists an element $u \in U_{\mathcal{H}}$ such that $h(u) \neq h(u')$. As a result,

$$q_h((u,0)) = h(u) \neq h(u') = q_{h'}((u,0)).$$

Therefore, U is a separator set for $\mathcal{Q}_{\mathcal{H}}$. □

Thus, if a hypothesis class $\mathcal{H}$ has a separator set of size m, so does the set of queries $\mathcal{Q}_{\mathcal{H}}$ representing the empirical loss of the hypotheses $h \in \mathcal{H}$.

Separator Sets for Common Hypothesis Classes.

In this section we provide some examples of hypothesis classes with small separator sets. We say that a class of queries $\mathcal{Q}$ is self-dual of $\mathcal{Q} = \mathcal{Q}_{\text{dual}}$.

Conjunctions, Disjunctions, and Parities.

We begin with some easy but important cases that can be verified by inspection:

Fact A.2.2. *Let $\mathcal{X}_A = \{0,1\}^d$. For every $j \in [d]$, let $e_j \in \mathcal{X}_A$ be a boolean vector that has 1 in the j-th coordinate and 0 in all others, and let $\overline{e}_j \in \mathcal{X}_A$ be the vector that has 0 in the j-th coordinate and 1 in all others. Let $U = \{e_j \mid j \in [d]\}$ and $\overline{U} = \{\overline{e}_j \mid j \in [d]\}$. Then for the following hypothesis classes:*

- *$\overline{U}$ is a separator set for conjunctions over $\mathcal{X}_A$: $\{\wedge_{j \in S} x_j \mid S \subseteq [d]\}$;*
- *U is a separator set for disjunctions over $\mathcal{X}_A$: $\{\vee_{j \in S} x_j \mid S \subseteq [d]\}$;*
- *U is a separator set for parities over $\mathcal{X}_A$: $\{\oplus_{j \in S} x_j \mid S \subseteq [d]\}$.*

Hence, each of these classes has a separator set of size d, equal to the data dimension. Moreover, each of these classes is self-dual.

Remark A.2.3. *Note here we have defined* monotone *conjunctions, disjunctions, and parities — i.e. in which the literals cannot appear negated. Up to a factor of 2 in the dimension, this is without loss of generality, since we can add d extra coordinates to each example which by convention will encode the negation of each of the values in the first d coordinates. This allows us to handle non-monotone conjunctions, disjunctions, and parities as well.*

Discrete Halfspaces.

Discrete halfspaces are a richer set of hypotheses that generalize both conjunctions and disjunctions. Let $\mathcal{X}_A = B^d$ for some set $B \subseteq [-1,1]$. For example we could allow real valued

features by letting $B = [-1, 1]$, or we could take some discretization. We will assume that $0 \in B$. Halfspaces themselves will be defined with respect to vectors of weights w that are discretized to lie in some finite set $w \in \mathcal{V}^d$, for $\mathcal{V} \subseteq [-1, 1]$. Here we could take $|\mathcal{V}| = 2$ by requiring that the *weights* be defined over the hypercube ($\mathcal{V} = \{-1, 1\}$), or we could allow finer discretization. It is important that $\mathcal{V}$ be finite.

Definition A.2.4 (Halfspace Query)**.** *Given a weight vector $w \in \mathcal{V}^d$, the halfspace query parameterized by w is defined to be $q_w(x) = \mathbf{1}\{w \cdot x \geq 1\}$. Let $\mathcal{Q}_{\mathcal{V}} = \{q_w : w \in \mathcal{V}^d\}$.*

The value of "1" used as the intercept is arbitrary, and can is set without loss of generality at the cost of 1 extra data dimension.

Lemma A.2.5. *$\mathcal{Q}_{\mathcal{V}}$ has a separator set of size $(|\mathcal{V}| - 1)d$. In particular, if weights are defined over the hypercube ($\mathcal{V} = \{-1, 1\}$), then the separator set is of size d.*

Proof. Suppose the elements of $\mathcal{V}$ are $x_1 < x_2 < \ldots < x_{|\mathcal{V}|}$, which without loss of generality are distinct. We construct a separator set of size $(|\mathcal{V}| - 1)d$ as follows. Let $c_1, \ldots c_{|\mathcal{V}|-1}$ be a sequence such that c_v lies in $[\frac{1}{x_{v+1}}, \frac{1}{x_v})$. Define the vector $s_{jv} \in \mathcal{X}$ to take value c_v in coordinate j, and 0 elsewhere. We claim that $U = \{s_{jv}\}_{j=1,\ldots d, v=1\ldots|\mathcal{V}|-1}$ is a separator set for $\mathcal{Q}_{\mathcal{V}}$. Let $q_{w_1} \neq q_{w_2} \in \mathcal{Q}_{\mathcal{V}}$. Since $w_1 \neq w_2$ they must differ in some coordinate, call it k. Let $w_{1,k} = x_l, q_{2,k} = x_m$, and without loss of generality assume $x_m > x_l$. Then by construction of $\{c_v\}$ there exists c_v such that $c_v \geq \frac{1}{x_m}$, but $c_v < \frac{1}{x_l}$. We therefore have that $w_1 \cdot s_{kv} = x_l \cdot c_v < 1$, whereas $w_2 \cdot s_{kv} = x_m \cdot c_v \geq 1$, and hence $q_{w_1}(s_{kv}) = 0 \neq 1 = q_{w_2}(s_{kv})$. □

Finally, we note that the dual of $\mathcal{Q}_{\mathcal{V}}$ is $\mathcal{Q}_{\mathcal{B}}$ — and so if $B = \mathcal{V}$, the set of halfspace queries $\mathcal{Q}_{\mathcal{V}}$ is self-dual.

Decision Lists.

For simplicity, in this section we discuss *monotone* decision lists, in which variables cannot be negated — but as we have already remarked, this is without loss of generality up to a factor of 2 in the dimension. Here we define the general class of k-decision lists: 1-decision lists (often just referred to as decision lists) are a restricted class of binary decision tree in

which one child of every internal vertex must be a leaf. k-decision lists are a generalization in which each branching decision can depend on a conjunction of k variables.

Definition A.2.6. *A monotone k-decision list over $\mathcal{X}_A = \{0,1\}^d$ is defined by an ordered sequence $L = (c_1, b_1), \ldots, (c_l, b_l)$ and a bit b, in which each c_i is a monotone conjunction of at most k literals, and each $b_i \in \{0,1\}$. Given a pair (L, b), the decision list $q_{L,b}(x)$ computes as follows: it outputs $q_{L,b}(x) = b_j$ where j is the minimum index in L satisfying $c_j(a) = 1$. If c_j is the first conjunction that x satisfies in the definition of $q_{L,b}(x)$ we say that x binds at c_j. If no such index exists then $q_{L,b}(x) = b$, and we say $q_{L,b}(x)$ does not bind. k-decision lists are strict generalizations of k-DNF and k-CNF formulae.*

Lemma A.2.7. *The class of k-decision lists has a separator set of size $\leq \sum_{j=0}^{2k} \binom{d}{j}$. In particular, 1-decision lists have a separator set of size $O(d^2)$.*

Proof. Let MC_k denote the set of monotone conjunctions of $\leq k$ literals over $\mathcal{X}_A$. For any two s, l in MC_k, let e_{sl} denote the element $a \in \mathcal{X}_A$ such that all literals appearing in either s or l are set to 1, and all others are set to 0. Define $U = \{e_{sl} : s, l \in MC_k\}$. Since each e_{sl} corresponds to setting between 0 and $2k$ of the d variables to 1, there are $\sum_{j=0}^{2k} \binom{d}{j}$ elements in U.

Now let $(L_1, b_1) \neq (L_2, b_2)$ be two distinct k-decision lists. Since the two decision lists are distinct there must exist $x \in \mathcal{X}_A$ such that $q_{(L_1,b_1)}(x) \neq q_{(L_2,b_2)}(x)$. Let c_1, c_2 be the conjunctions on which $(L_1, b_1), (L_2, b_2)$ bind on x respectively. (If L_i does not bind on x, set c_i to be the empty conjunction). Define $e_{c_1,c_2} \in U$ as above. We claim that $q_{(L_1,b_1)}(e_{c_1,c_2}) = q_{(L_1,b_1)}(x) \neq q_{(L_2,b_2)}(x) = q_{(L_2,b_2)}(e_{c_1,c_2})$, and hence $e_{c_1,c_2} \in U$ distinguishes (L_1, b_1) from (L_2, b_2), which proves the claim. The key fact that $q_{(L_i,b_i)}(e_{c_1,c_2}) = q_{(L_i,b_i)}(x)$ follows from the fact that (L_i, b_i) still binds at c_i (or does not bind at all) on input x, and it can't bind earlier since any monotone conjunction satisfied by e_{c_1,c_2} is satisfied by x. □

Other Classes of Functions.

In this section, we have exhibited simple constructions of small universal identification sets for conjunctions, disjunctions, parities, discrete halfspaces, and k-decision lists. This is

not an exhaustive enumeration of such classes — we give these as examples of the most frequently studied classes of boolean functions in the PAC learning literature. However, short universal identification sets for other classes of functions are known. For example:

Theorem A.2.8 (Goldman et al. (1993))**.** *There exist polynomially sized universal identification sets for the following two classes of circuits:*

1. *Logarithmic depth read-once majority formulas, and*
2. *Logarithmic depth read-once positive NAND formulas.*

A.2.2. RSPM with Gaussian Perturbations

We now present a Gaussian variant of the RSPM algorithm.

Gaussian RSPM
Given: A separator set $U = \{e_1, \ldots, e_m\}$ for a class of statistical queries $\mathcal{Q}$ and a weighted optimization oracle $\mathcal{O}^*$ for $\mathcal{Q}$, privacy parameters ε and $\delta \in (0, 1/e)$, and $\sigma = \frac{7\sqrt{m \ln(1/\delta)}}{\varepsilon}$.
Input: A dataset $S \in \mathcal{X}^n$ of size n.
Output: A statistical query $q \in \mathcal{Q}$.

Sample independently $\eta_i \sim \mathcal{N}(0, \sigma^2)$ for $i \in \{1, \ldots, m\}$
Construct a weighted dataset WD of size $n + m$ as follows:

$$WD(S, \eta) = \{(x_i, 1) : x_i \in S\} \cup \{(e_i, \eta_i) : e_i \in U\}$$

Output $q = \mathcal{O}(WD(S, \eta))$.

Theorem A.2.9 (Utility)**.** *The Gaussian RSPM algorithm is an oracle-efficient (α, β)-minimizer for $\mathcal{Q}$ for:*

$$\alpha = O\left(\frac{m\sqrt{m \ln(2m/\beta) \ln(1/\delta)}}{\varepsilon n}\right)$$

Proof. Note that for each noise variable η_i, we have the following tail bound:

$$\Pr[|\eta_i| \geq t] \leq 2 \exp\left(-\frac{t^2}{2\sigma^2}\right)$$

Taking a union bound, we have with probability at least $1-\beta$ that

$$\max_i |\eta_i| \leq \sqrt{2\ln(2m/\beta)}\sigma$$

Then the proof follows from the same reasoning in the proof for Theorem 4.3.2. □

Theorem A.2.10 (Privacy). *If $\mathcal{O}^*$ is a weighted optimization oracle for $\mathcal{Q}$, then the Gaussian Report Separator-Perturbed Min algorithm is (ε, δ)-differentially private.*

Proof. In the following, we will inherit the notation from Section 4.3. We will denote the output by the algorithm on input dataset S under realizations of the perturbations η as $\mathcal{Q}(S,\eta) = \mathcal{O}^*(WD(S,\eta))$. For any $q \in \mathcal{Q}$, let $\mathcal{E}(q,S) = \{\eta : \mathcal{Q}(S,\eta) = q\}$. We will use the same mapping $f_q(\eta) : \mathbb{R}^m \to \mathbb{R}^m$ as defined in Section 4.3. We will write P_G to denote the pdf of the distribution $\mathcal{N}(0,\sigma^2)$. We will also again define the set B as the set of η for which there are multiple minimizers $\hat{q}$. For every $\eta \in \mathbb{R}^m \setminus B$, let $\hat{q}_\eta$ be the unique minimizer. Note that Lemmas 4.3.5 and 4.3.7 both hold in our setting, and in particular, Lemma 4.3.7 holds because of the continuity of the Gaussian distribution.

Similar to the standard analysis for the Gaussian mechanism Dwork and Roth (2014b), we will leverage the fact that the distribution $\mathcal{N}(0,\sigma^2 I)$ is independent of the orthonormal basis from which its constituent normals are drawn, so we have the freedom to choose the underlying basis without changing the distribution. For any $r \in \mathbb{R}^m$ and any $q \in \mathcal{Q}$, fix such a basis $b_1,\ldots,b_m$ such that b_1 is parallel to $v = f_q(r) - r$. A random draw η from $\mathcal{N}(0,\sigma^2 I)$ can be realized by the following process: first draw signed lengths $\lambda_i \sim \mathcal{N}(0,\sigma^2)$, for $i \in [m]$, then define $\eta^{[i]} = \lambda_i b_i$, and finally let $\eta = \sum_{i=1}^m \eta^{[i]}$. For each $i \in [m]$, let $r^{[i]}$ be the projection of r onto the direction of b_i, which gives $r = \sum_{i=1}^m r^{[i]}$.

Lemma A.2.11. *Suppose that $\sigma > 1$. For any $r \in \mathbb{R}^m$, $q \in \mathcal{Q}$,*

$$P_G(r) \leq \exp\left(\frac{1}{2\sigma^2}\left(m + 2\sqrt{m}\|r^{[1]}\|_2\right)\right) P_G(f_q(r)).$$

Proof. Note that for any $r \in \mathbb{R}^m$, we have

$$P_G(r) = \frac{1}{(2\pi)^{m/2}\sigma^m} \exp\left(-\frac{\|r\|_2^2}{2\sigma^2}\right).$$

We will write $\nu = f_q(r) - r$. It follows that

$$\frac{P_G(r)}{P_G(f_q(r))} = \exp\left(\frac{1}{2\sigma^2}\left(\|r+\nu\|_2^2 - \|r\|_2^2\right)\right)$$

Now we can write

$$\|r+\nu\|_2^2 = \|\nu + r^{[1]}\|_2^2 + \sum_{i=1}^{m} \|r^{[i]}\|_2^2, \qquad \|r\|_2^2 = \sum_{i=1}^{m} \|r^{[i]}\|_2^2.$$

It follows that

$$\|r^{[1]}+\nu\|_2^2 - \|r^{[1]}\|_2^2 = \|\nu\|_2^2 + 2\|\nu\|_2\|r^{[1]}\|_2 \le m + 2\sqrt{m}\|r^{[1]}\|_2$$

This means

$$\frac{P_G(r)}{P_G(f_q(r))} \le \exp\left(\frac{1}{2\sigma^2}\left(m + 2\sqrt{m}\|r^{[1]}\|_2\right)\right),$$

which completes the proof. □

To finish up the privacy analysis, note that by Lemma A.2.11, the ratio $P_G(\eta)/P_G(f_q(\eta))$ is bounded by $\exp(\varepsilon)$, as long as $\|\eta^{[1]}\|_2 < \sigma^2\varepsilon/\sqrt{m} - \sqrt{m}/2$. Now we will bound the probability that the random vector $\eta^{[1]}$ has norm exceeding this bound. First, observe that $\|\eta^{[1]}\|_2 = |\lambda_1|$, where λ_1 is a random draw from the distribution $\mathcal{N}(0,\sigma^2)$. Since λ_1^2 is a χ^2 random variable with degree of freedom 1, we can apply the following tail bound Laurent and Massart (2000): for any $t > 0$,

$$\Pr[\lambda_1^2 \ge \sigma^2\left(\sqrt{2t}+1\right)^2] \le \exp(-t)$$

which can be further simplied to

$$\Pr[\|\eta^{[1]}\|_2 \geq \sigma\left(\sqrt{2t}+1\right)] \leq \exp(-t)$$

In other words, for any $\delta \in (0, 1/e)$, with probability at least $1-\delta$, we have

$$\|\eta^{[1]}\|_2 < \sigma(\sqrt{2\ln(1/\delta)}+1) \equiv \Lambda$$

It follows that $\Lambda \leq \sigma^2\varepsilon/\sqrt{m} - \sqrt{m}/2$, as long as $\sigma = \frac{c\sqrt{m\ln(1/\delta)}}{\varepsilon}$ for any $c \geq 3.5$. We will use this value of σ for the remainder of the analysis. Now let $L = \{\eta \in \mathbb{R}^m \mid \|\eta^{[1]}\|_2 > \Lambda\}$, then we know that $\Pr[\eta \in L] < \delta$. Let $S \subset \mathcal{Q}$ be a subset of queries. It follows that:

$$\begin{aligned}
\Pr\left[\eta \in \bigcup_{\hat{q}\in S} \mathcal{E}(\hat{q}, S)\right] &= \int_{\mathbb{R}^m} P_G(\eta)\mathbb{1}\left(\eta \in \bigcup_{\hat{q}\in S} \mathcal{E}(\hat{q}, S)\right) d\eta \\
&= \int_{(\mathbb{R}^m\setminus B)\setminus L} P_G(\eta)\mathbb{1}\left(\eta \in \bigcup_{\hat{q}\in S} \mathcal{E}(\hat{q}, S)\right) d\eta + \int_L P_G(\eta)\mathbb{1}\left(\eta \in \bigcup_{\hat{q}\in S} \mathcal{E}(\hat{q}, S)\right) d\eta \\
&\leq \int_{(\mathbb{R}^m\setminus B)\setminus L} P_G(\eta)\mathbb{1}\left(\eta \in \bigcup_{\hat{q}\in S} \mathcal{E}(\hat{q}, S)\right) d\eta + \delta \\
&= \sum_{\hat{q}\in S} \int_{\mathbb{R}^m\setminus(B\cup L)} P_G(\eta)\mathbb{1}\left(\eta \in \mathcal{E}(\hat{q}, S)\right) d\eta + \delta \\
&\leq \sum_{\hat{q}\in S} \int_{\mathbb{R}^m\setminus(B\cup L)} P_G(\eta)\mathbb{1}\left(f_{\hat{q}}(\eta) \in \mathcal{E}(\hat{q}, S')\right) d\eta + \delta (\textit{Lemma 4.3.5}) \\
&\leq \sum_{\hat{q}\in S} \int_{\mathbb{R}^m\setminus(B\cup L)} \exp(\varepsilon) P_G(f_{\hat{q}}(\eta))\mathbb{1}\left(f_{\hat{q}}(\eta) \in \mathcal{E}(\hat{q}, S')\right) d\eta + \delta (\textit{Lemma A.2.11}) \\
&= \sum_{\hat{q}\in S} \int_{\mathbb{R}^m\setminus(f_{\hat{q}}(B)\cup f_{\hat{q}}(L))} \exp(\varepsilon) P_G(\eta)\mathbb{1}\left(\eta \in \mathcal{E}(\hat{q}, S')\right) \left|\frac{\partial f_{\hat{q}}}{\partial \eta}\right| d\eta + \delta \\
&\leq \exp(\varepsilon) \sum_{\hat{q}\in S} \int_{\mathbb{R}^m} P_G(\eta)\mathbb{1}\left(\eta \in \mathcal{E}(\hat{q}, S')\right) d\eta + \delta \quad \text{since } \left(\forall \hat{q}, \left|\frac{\partial f_{\hat{q}}}{\partial \eta}\right| = 1\right) \\
&= \exp(\varepsilon) \Pr\left[\eta \in \bigcup_{\hat{q}\in S} \mathcal{E}(\hat{q}, S')\right] + \delta
\end{aligned}$$

This completes the proof. □

A.2.3. Proofs and Details for Theorem 4.3.9

Lemma 2.2.6. *Let $\mathcal{A}_{\mathcal{O}}$ be a certifiable-oracle dependent algorithm that is oracle equivalent to $\mathcal{A}$. Then for any fixed input dataset S, there exists a coupling between $\mathcal{A}(S)$ and $\mathcal{A}_{\mathcal{O}}(S)$ such that* $\Pr[\mathcal{A}_{\mathcal{O}}(S) = a | \mathcal{A}_{\mathcal{O}}(S) \neq \perp] = \Pr[\mathcal{A}(S) = a | \mathcal{A}_{\mathcal{O}}(S) \neq \perp]$.

Proof. We can assume without loss of generality that $\mathcal{A}_{\mathcal{O}}$ and $\mathcal{A}$ draw all their randomness up front in the form of a random seed η, and are then a deterministic function of the random seed and the input dataset. By definition, during the run of $\mathcal{A}_{\mathcal{O}}$, the algorithm generates a (possibly randomized, and possibly adaptively chosen) sequence of inputs to the optimization oracle $\mathcal{O}$, $\{wd_1, \ldots wd_m\}$, where each wd_i is a weighted dataset. We denote the output of the i^{th} optimization problem by o_i. After the m^{th} optimization problem, $\mathcal{A}_{\mathcal{O}}$ outputs a deterministic outcome $a = h(o_1, o_2, \ldots o_m)$. Given access to a perfect optimization oracle $\mathcal{O}^*$, $\mathcal{A}_{\mathcal{O}^*}$ is simply $\mathcal{A}$ – this is the definition of oracle equivalence. We construct a coupling between an algorithm $\mathcal{M}$ and $\mathcal{A}_{\mathcal{O}}$, and then argue running $\mathcal{M}$ is the same as running $\mathcal{A}$:

Input: A dataset S, random seed η, heuristic oracle $\mathcal{O}$, perfect oracle $\mathcal{O}^*$.
Output: values $M(S, \eta), \mathcal{A}(S, \eta)$

```
Run A_O(η, S) - generating the first optimization problem wd_1.
for i = 1 ... m do
    Compute o_i = O(wd_i)
    if o_i = ⊥ then
        Output A_O(S, w) = ⊥
        Set o_i = O*(wd_i)
    Generate wd_{i+1} adaptively as a function of previous outputs (o_i, o_{i-1}, ... o_1)
Output M(S, η) = h(o_1, ... o_m) = a
if A_O(S, η) = ⊥ has not been output then
    Output A_O(S, w) = a
```

The procedure starts by generating a random seed η and initializing a run of $\mathcal{A}_{\mathcal{O}}(\eta, S)$ - generating the first optimization problem wd_1. If the oracle $\mathcal{O}$ fails on input $wd_1, \mathcal{A}_{\mathcal{O}}$ outputs $\perp$. In this case the next optimization input wd_2 is generated as a function of the output of the perfect oracle $\mathcal{O}^*(wd_1)$. If it succeeds, we simply generate the next output

as a function of $\mathcal{O}(wd_1)$ (which is the same as $\mathcal{O}^*(wd_1)$ by definition of certifiability). This process continues until we solve the m^{th} optimization problem, and output $M(\eta, S), \mathcal{A}_{\mathcal{O}}(\eta, S)$ as described above. Now it is clear that if the oracle doesn't fail, we generate the same output a for $\mathcal{A}_{\mathcal{O}}$ and $\mathcal{M}$. No matter whether or not $\mathcal{O}$ fails, $\mathcal{M}$ has output that corresponds to perfectly solving the optimization problems generated with input S, η, and so it is equivalent to running $\mathcal{A}$. Moreover, whenever the oracle does not fail, $\mathcal{A}$ and $\mathcal{M}$ have the same output, by construction. This completes the proof. □

Definition A.2.12 (PEMANTLE and PERES (2014))**.** *Let $(X_1, \ldots, X_n) \subset \{0,1\}^n$ be an ensemble of n $\{0,1\}$-valued random variables. We say that $(X_1, \ldots X_n)$ satisfy the stochastic covering property, if for any $I \subset [n]$, $J = [n] \backslash I$, and $a \geq a' \in \{0,1\}^{|I|}$, where $\geq$ denotes coordinate-wise dominance, such that $\|a' - a\|_1 = 1$, there is a coupling v of the distributions μ, μ' on $(X_k)_{k \in J}$ conditioned on $(X_k)_{k \in I} = a$ or $(X_k)_{k \in I} = a'$ respectively, such that $v(x, y) = 0$ unless $x \leq y$ and $\|x - y\|_1 \leq 1$.*

Lemma A.2.13. *Given a set $|S| = n$, subsample $k \leq n$ elements without replacement. Let $X_i \in \{0,1\}$ be 1 if element $x_i \in S$ is subsampled, else 0. Then $(X_1, \ldots X_n)$ satisfy the stochastic covering property.*

Proof. For $a \in \{0,1\}^{|I|}$ let $|a|$ be the number of 1's in a. Then the distribution of $x = X_J | a$ corresponds to subsampling $k - |a|$ elements from $(x_k)_{k \in J}$, and the distribution of $y = X_J | a'$ corresponds to subsampling $k - |a| + 1$ elements from $(x_k)_{k \in J}$ without replacement. To establish the stochastic covering property, we exhibit a coupling v of x, y:

To generate y subsample $k - |a| + 1$ elements from $(x_k)_{k \in J}$ without replacement. Let x be the first $k - |a|$ such elements subsampled. Both x, y constructed as such have the correct marginal distributions, and by construction $x \leq y, \|x - y\|_1 = 1$ always. □

Definition A.2.14. *$(X_1, \ldots, X_n) \subset \{0,1\}^n$ are k-homogenous if $\Pr[\sum_{i=1}^n X_i = k] = 1$.*

Theorem A.2.15 (Theorem 3.1 in PEMANTLE and PERES (2014))**.** *Let $(X_1, \ldots X_n) \in \{0,1\}$ be k-homogenous random variables satisfying the stochastic covering property. Let $f : \{0,1\}^n \to \mathbb{R}$*

be an ε–Lipschitz function, and let $\mu = \mathbb{E}[f(X_1, \ldots X_n)]$. Then for any $t > 0$:

$$\Pr[|f(X_1, \ldots X_n) - \mu| \geq t] \leq 2e^{\frac{-t^2}{8\varepsilon^2 k}}$$

Lemma 4.3.10. *Let $\mathcal{P}(S) \sim \mathcal{P}^S_{split}$. Let $\mathcal{A} : \mathcal{X}^l \to \mathcal{M}$ be an $(\varepsilon', 0)$ differentially private algorithm, where: $\varepsilon' = \frac{1}{\sqrt{8\frac{n}{K}\log(2K/\delta)}}$. Fix $\Omega \subset M$, and let $q_\Omega(S_i) = \log \Pr[\mathcal{A}(S_i) \in \Omega]$. Define Q to be the event*

$$Q = \{\mathcal{P}(S) : \max_{i,j \in 1 \ldots K} |q_\Omega(S_i) - q_\Omega(S_j)| \leq 2\}.$$

Then over the random draw of $\mathcal{P}(S) \sim \mathcal{P}^S_{split}$, $\Pr[Q] \geq 1 - \delta$.

Proof. Fix any index k of the partition. Let $\{X_i\}_{i=1}^n$ be the indicator random random variables indicating that element i in S, is included in S_k. Since S_k is entirely determined by $\{X_i\}$, we can write $q_\Omega(S_k)$ as a function of $\{X_i\}$, e.g. $q_\Omega(X_1, \ldots X_n)$. Moreover, by definition of ε-differential privacy, q_Ω is ε-Lipschitz, i.e. for any X_i, X_i':

$$|q_\Omega(X_1, \ldots X_i, \ldots X_n) - q_\Omega(X_1, \ldots X_i', \ldots X_n)| \leq \varepsilon$$

By Lemma A.2.13 proven in the Appendix, $(X_1, \ldots X_n)$ satisfy what is called the *stochastic covering property*, a type of negative dependence. Since $|S_k| = n/K$, $(X_1, \ldots X_n)$ are n/K-homogenous. Thus by Theorem 1 of PEMANTLE and PERES (2014), with probability $1 - \delta/K$:

$$|q_\Omega(S_k) - \mathop{\mathbb{E}}_{S_k \sim \mathcal{P}^S_{split}}[q_\Omega(S_k)]| \leq \varepsilon' \sqrt{8\frac{n}{K}\log(2K/\delta)} = 1$$

So by a union bound this holds for all $k = 1 \ldots K$ with probability at least $1 - \delta$. Since for all $i, j, \mathbb{E}_{S_i \sim \mathcal{P}^S_{split}}[q_\Omega(S_i)] = \mathbb{E}_{S_j \sim \mathcal{P}^S_{split}}\left[q_\Omega(S_j)\right]$, by the triangle inequality for all $i, j, |q_\Omega(S_i) - q_\Omega(S_j)| \leq 2$ with probability $1 - \delta$, as desired.

□

Lemma 4.3.11.

$$\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \leq \frac{\sum_{\mathcal{P}(S) \in S_Q, \boldsymbol{o} \in \mathcal{F}} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \boldsymbol{o}, \mathcal{L}] \Pr[\boldsymbol{o}, P(S) | \mathcal{L}] + 4\delta}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]}$$

Proof. Conditioning on $\mathcal{L}$ and using $\Pr[\mathcal{L}] \geq 1 - 2\delta$ we have:

$$\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) = a]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \leq \frac{2\delta + \Pr\left[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | \mathcal{L}\right]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]}$$

Expanding $\Pr\left[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | \mathcal{L}\right]$ by conditioning on $\mathcal{F}, \mathcal{P}(S)$ and using the law of total probability we have:

$$\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) = a | \mathcal{L}]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} = \frac{\sum_{\mathcal{P}(S), \mathbf{o} \in \mathcal{F}} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S) | \mathcal{L}]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} = \tag{A.2}$$

Separating the summation in the numerator over $\mathcal{P}(S)$ into S_Q, S_Q^c we have:

$$\sum_{\mathcal{P}(S), \mathbf{o} \in \mathcal{F}} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S) | \mathcal{L}] =$$

$$\sum_{\mathcal{P}(S) \in S_Q, \mathbf{o} \in \mathcal{F}} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S) | \mathcal{L}] +$$

$$\sum_{\mathcal{P}(S) \in S_Q^c, \mathbf{o} \in \mathcal{F}} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S) | \mathcal{L}] \tag{A.3}$$

Rewriting the second term,

$$\sum_{\mathcal{P}(S) \in S_Q^c, \mathbf{o} \in \mathcal{F}} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S) | \mathcal{L}] =$$

$$\sum_{\mathcal{P}(S) \in S_Q^c} (\sum_{\mathbf{o} \in \mathcal{F}} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, | P(S), \mathcal{L}]) \Pr[P(S) | \mathcal{L}] =$$

$$\sum_{\mathcal{P}(S)\in S_Q^c} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathcal{L}]) \Pr[P(S)|\mathcal{L}] \leq \sum_{\mathcal{P}(S)\in S_Q^c} \Pr[P(S)|\mathcal{L}] = \Pr[Q^c|\mathcal{L}]$$

The first equality follows from the fact that $\Pr[\mathbf{o}, \mathcal{P}(S)|\mathcal{L}] = \Pr[\mathbf{o}|\mathcal{P}(S), \mathcal{L}] \Pr[\mathcal{P}(S)|\mathcal{L}]$, and the second equality follows from the law of total probability. Since $\Pr[Q^c] \leq \delta$, and $\Pr[\mathcal{L}] \geq 1-2\delta$, $\Pr[Q^c|\mathcal{L}] \leq \delta/(1-2\delta) \leq 2\delta$, for $\delta \leq 1/4$. Thus

$$\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) = a|\mathcal{L}]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \leq \frac{\sum_{\mathcal{P}(S)\in S_Q, \mathbf{o}\in\mathcal{F}} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S)|\mathcal{L}] + 2\delta}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]}$$

Combining this bound, with the bound $\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S})=a]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}')\in\Omega]} \leq \frac{2\delta+\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S})\in\Omega|\mathcal{L}]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}')\in\Omega]}$, establishes the result.

□

Lemma 4.3.12. *Fix any $\boldsymbol{o}$, any $\mathcal{P}(S) \in S_Q$, and index $j \in I^{\boldsymbol{o}}_{pass}$, i.e. $o_j \neq \perp$. Then:*

$$\Pr[\mathcal{A}_{\mathcal{O}}(S_j) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_j) \neq \perp, \mathcal{L}] \leq \frac{e^2}{(1-\delta)^2} \frac{1}{|\boldsymbol{o}|-1} \sum_{i\in I^{\boldsymbol{o}}_{pass}, i\neq j} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] + \frac{\delta e^2}{(1-\delta)}$$

Proof of Lemma 4.3.12. By Equation 4.3, we know:

$$\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] = \frac{1}{|\mathbf{o}|} \sum_{i\in I^{\mathbf{o}}_{pass}} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) = a | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp]$$

We also know that since we've conditioned on $\mathcal{L}$ (and hence on E), for each $i \in I^{\mathbf{o}}_{pass}$ on the RHS of the above equation, $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) = \perp] \leq \delta$. By Lemma 2.2.6, we know that there exists a coupling between $\mathcal{A}_{\mathcal{O}}(S_i)$ and $\mathcal{A}(S_i)$ such that $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) = a | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] = \Pr[\mathcal{A}(S_i) = a | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp]$.

By the law of total probability $\Pr[\mathcal{A}(S_i) = a] =$

$$\Pr[\mathcal{A}(S_i) = a | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] + \Pr[\mathcal{A}(S_i) = a | \mathcal{A}_{\mathcal{O}}(S_i) = \perp] \Pr[\mathcal{A}_{\mathcal{O}}(S_i) = \perp]$$

Then $\Pr[\mathcal{A}(S_i) = a | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \leq \Pr[\mathcal{A}(S_i) = a] \implies \Pr[\mathcal{A}(S_i) = a | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \leq \frac{1}{1-\delta} \Pr[\mathcal{A}(S_i) = a]$, and similarly $\Pr[\mathcal{A}(S_i) = a | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \geq \Pr[\mathcal{A}(S_i) = a] - \delta$.

Since we've shown that each of the conditional probabilities $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) = a | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp]$ is close to $\Pr[\mathcal{A}_{\mathcal{O}}(S_i) = a]$, since we assume $\mathcal{P}(S) \in S_Q$, we know they are close to each other. Using the inequalities above:

$$\Pr[\mathcal{A}(S_j) = a | o_j \neq \perp] \leq \frac{1}{1-\delta} \Pr[\mathcal{A}(S_j) = a] \leq \frac{e^2}{1-\delta} \Pr[\mathcal{A}(S_i) = a] \leq \frac{e^2}{1-\delta} \left(\frac{\Pr[\mathcal{A}(S_i) = a | o_i \neq \perp]}{1-\delta} + \delta \right), \quad \text{(A.4)}$$

where the middle inequality follows from the definition of S_Q. Summing both sides over $i : o_i \neq \perp, i \neq j$ and rearranging gives the desired result. □

Lemma 4.3.13.

$$\sum_{\mathcal{P}(S) \in S_Q, \boldsymbol{o} \in \mathcal{F}} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \boldsymbol{o}, \mathcal{L}] \Pr[\boldsymbol{o}, P(S) | \mathcal{L}] \leq$$

$$\left(1 + \frac{e^2}{(1-\delta)^2 (\frac{1}{\varepsilon} - 1)}\right) \sum_{P(S) \in S_Q} \left(\sum_{\boldsymbol{o} \in \mathcal{F}} \left(\frac{1}{|\boldsymbol{o}|} \sum_{i \in I^{\boldsymbol{o}}_{pass}, i \neq 1} \Pr[\mathcal{A}_{\mathcal{O}}(S_i) \in \Omega | \mathcal{A}_{\mathcal{O}}(S_i) \neq \perp] \right) \Pr[\boldsymbol{o} | P(S)] \right) \Pr[P(S)] + \frac{\varepsilon \delta e^2}{1-\delta} \quad \text{(4.5)}$$

Proof. Denote $\sum_{P(S)\in S_Q}\Big(\sum_{\mathbf{o}\in\mathcal{F}:o_1\neq\perp}\big(\frac{1}{|\mathbf{o}|}\sum_{i\in I^{\mathbf{o}}_{pass}}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)=a|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp]\big)\Pr[\mathbf{o}|P(S)]$, by $(\star)$. By Lemma 4.3.12, $(\star)\leq$

$$\sum_{P(S)\in S_Q}\Big(\sum_{\mathbf{o}\in\mathcal{F}:o_1\neq\perp}\big(\frac{\delta e^2}{(1-\delta)|\mathbf{o}|}+(1+\frac{e^2}{(|\mathbf{o}|-1)(1-\delta)^2})\cdot\frac{1}{|\mathbf{o}|}\cdot \sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)=a|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp,i^*=i]\big)\Pr[\mathbf{o}|P(S)]\Big)\Pr[P(S)] \quad (A.5)$$

Pulling out the $\frac{\delta e^2}{(1-\delta)|\mathbf{o}|}$, we see that $\sum_{\mathbf{o}\in\mathcal{F}:o_1\neq\perp}\frac{\delta e^2}{(1-\delta)|\mathbf{o}|}\Pr[\mathbf{o}|P(S)]\leq\sum_{\mathbf{o}\in\mathcal{F}}\frac{\delta e^2}{(1-\delta)}\cdot\varepsilon\Pr[\mathbf{o}|P(S)]=\frac{\varepsilon\delta e^2}{(1-\delta)}$. Here we've used the fact that $|\mathbf{o}|>\frac{1}{\varepsilon}$. Similarly, $\sum_{P(S)\in S_Q}\frac{\varepsilon\delta e^2}{(1-\delta)}\Pr[P(S)]\leq\frac{\varepsilon\delta e^2}{(1-\delta)}$. Applying to $(\star)$, we get $(\star)\leq$:

$$\sum_{P(S)\in S_Q}\Big(\sum_{\mathbf{o}\in\mathcal{F}:o_1\neq\perp}\big((1+\frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)})\cdot\frac{1}{|\mathbf{o}|}\cdot \sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)=a|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp,i^*=i]\big)\Pr[\mathbf{o}|P(S)]\Big)\Pr[P(S)]+\frac{\varepsilon\delta e^2}{1-\delta} \quad (A.6)$$

Applying this upper bound on $(\star)$ gives:

$$\sum_{\mathcal{P}(S)\in S_Q,\mathbf{o}\in\mathcal{F}}\Pr[\mathcal{A}^{prsm}_{\mathcal{O}}(\mathcal{S})\in\Omega|P(S),\mathbf{o},\mathcal{L}]\Pr[\mathbf{o},P(S)|\mathcal{L}]\leq$$

$$\sum_{P(S)\in S_Q}\Big(\big(\sum_{\mathbf{o}\in\mathcal{F}:o_1\neq\perp}\big((1+\frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)})\cdot\frac{1}{|\mathbf{o}|}\sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)=a|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp,i^*=i]\big) \Pr[\mathbf{o}|P(S)]\big)+ \quad (A.7)$$

$$\sum_{\mathbf{o}\in\mathcal{F}:o_1=\perp}\big(\frac{1}{|\mathbf{o}|}\sum_{i\in I^{\mathbf{o}}_{pass}}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)=a|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp,i^*=i]\Pr[\mathbf{o}|P(S)]\big)\Big)\Pr[P(S)]\leq$$

$$(1+\frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)})\cdot$$
$$\sum_{P(S)\in S_Q}\Bigg(\sum_{\mathbf{o}\in\mathcal{F}}\Big(\frac{1}{|\mathbf{o}|}\sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)=a|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp,i^*=i]\Big)\Pr[\mathbf{o}|P(S)]\Bigg)\Pr[P(S)] \quad (A.8)$$

□

End of proof of Theorem 4.3.9.

Proof. First we rewrite the numerator:

$$\sum_{\mathbf{o}\in\mathcal{F}}\Big(\frac{1}{|\mathbf{o}|}\sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)\in\Omega|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp]\Pr[\mathbf{o}|P(S)]\Big)=$$

$$\sum_{\mathbf{o}_{-1}\in\mathcal{F}}\sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}\Bigg(\Big(\frac{1}{|\mathbf{o}_{-1}|}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)\in\Omega|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp]\Pr[o_1=\perp,\mathbf{o}_{-1}|P(S)]+$$

$$\frac{1}{|\mathbf{o}_{-1}+1|}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)\in\Omega|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp]\Pr[o_1=1,\mathbf{o}_{-1}|P(S)]\Big)\Bigg)\leq$$

$$\sum_{\mathbf{o}_{-1}\in\mathcal{F}}\sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}\Big(\frac{1}{|\mathbf{o}_{-1}|-1}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)\in\Omega|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp]\Pr[\mathbf{o}_{-1}|P(S)]\Big),$$

where we've used the fact that $\Pr[o_1=\perp,\mathbf{o}_{-1}|P(S)]+\Pr[o_1\neq\perp,\mathbf{o}_{-1}|P(S)]=\Pr[\mathbf{o}_{-1}|P(S)]$. Similarly, we can use this same trick to lower bound the denominator:

$$\sum_{\mathbf{o}\in\mathcal{F}}\sum_{i\in I^{\mathbf{o}}_{pass}}(\frac{1}{|\mathbf{o}|}\Pr[\mathcal{A}_{\mathcal{O}}(S'_i)\in\Omega|\mathcal{A}_{\mathcal{O}}(S'_i)\neq\perp]\Pr[\mathbf{o}|P(S')])\geq$$

$$\sum_{\mathbf{o}_{-1}\in\mathcal{F}}\sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}(\frac{1}{|\mathbf{o}_{-1}|}\Pr[\mathcal{A}_{\mathcal{O}}(S'_i)\in\Omega|\mathcal{A}_{\mathcal{O}}(S'_i)\neq\perp]\Pr[o_1\neq\perp,\mathbf{o}_{-1}|P(S')]+$$

$$\frac{1}{|\mathbf{o}_{-1}|-1}\Pr[\mathcal{A}_{\mathcal{O}}(S_i')\in\Omega|\mathcal{A}_{\mathcal{O}}(S_i')\neq\perp]\Pr[o_1=\perp,\mathbf{o}_{-1}|P(S')])\geq$$

$$\sum_{\mathbf{o}_{-1}\in\mathcal{F}}\sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}\Big(\frac{1}{|\mathbf{o}_{-1}|}\Pr[\mathcal{A}_{\mathcal{O}}(S_i')\in\Omega|\mathcal{A}_{\mathcal{O}}(S_i')\neq\perp]\Pr[\mathbf{o}_{-1}|P(S')]\Big)$$

Substituting these inequalities into (4.7), we get that (4.7) $\leq$

$$\sup_{P(S)\sim P(S')}\frac{(1+\frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)})\sum_{\mathbf{o}_{-1}\in\mathcal{F}}\sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}\Big(\frac{1}{|\mathbf{o}_{-1}|-1}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)\in\Omega|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp]\Pr[\mathbf{o}_{-1}|P(S)]\Big)}{\sum_{\mathbf{o}_{-1}\in\mathcal{F}}\sum_{i\in I^{\mathbf{o}}_{pass},i\neq 1}\Big(\frac{1}{|\mathbf{o}_{-1}|}\Pr[\mathcal{A}_{\mathcal{O}}(S_i')\in\Omega|\mathcal{A}_{\mathcal{O}}(S_i')\neq\perp]\Pr[\mathbf{o}_{-1}|P(S')]\Big)}+$$

$$\frac{\frac{\varepsilon\delta e^2}{1-\delta}}{\Pr[\mathcal{A}^{prsm}_{\mathcal{O}}(\mathcal{S}')\in\Omega]}\leq \quad \text{(A.9)}$$

$$(1+\frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)})\cdot$$
$$\sup_{P(S)\sim P(S'),\mathbf{o}_{-1},i\neq 1}\frac{\frac{1}{|\mathbf{o}_{-1}|-1}\Pr[\mathcal{A}_{\mathcal{O}}(S_i)\in\Omega|\mathcal{A}_{\mathcal{O}}(S_i)\neq\perp]\Pr[\mathbf{o}_{-1}|P(S)]}{\frac{1}{|\mathbf{o}_{-1}|}\Pr[\mathcal{A}_{\mathcal{O}}(S_i')\in\Omega|\mathcal{A}_{\mathcal{O}}(S_i')\neq\perp]\Pr[\mathbf{o}_{-1}|P(S')]}+\frac{\frac{\varepsilon\delta e^2}{1-\delta}}{\Pr[\mathcal{A}^{prsm}_{\mathcal{O}}(\mathcal{S}')\in\Omega]} \quad \text{(A.10)}$$

Since $S_i=S_i'$ for all $i\neq 1$, this reduces to:

$$(1+\frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)})\cdot\sup_{\mathbf{o}_{-1}}\frac{|\mathbf{o}_{-1}|}{|\mathbf{o}_{-1}|-1}+\frac{\frac{\varepsilon\delta e^2}{1-\delta}}{\Pr[\mathcal{A}^{prsm}_{\mathcal{O}}(\mathcal{S}')\in\Omega]}\leq$$
$$(1+\frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)})\frac{1}{1-\varepsilon}+\frac{\frac{\varepsilon\delta e^2}{1-\delta}}{\Pr[\mathcal{A}^{prsm}_{\mathcal{O}}(\mathcal{S}')\in\Omega]} \quad \text{(A.11)}$$

since $|\mathbf{o}_{-1}|\geq\frac{1}{\varepsilon}$ by definition.

Following the chain of inequalities back to their genesis, we finally obtain:

$$\frac{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega | P(S), \mathbf{o}, \mathcal{L}] \Pr[\mathbf{o}, P(S) | \mathcal{L}]}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]} \leq (1 + \frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon} - 1)}) \frac{1}{1-\varepsilon} + \frac{\frac{\varepsilon\delta e^2}{1-\delta}}{\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega]},$$

which substituting into Lemma 4.3.11 gives:

$$\Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}) \in \Omega] \leq (1 + \frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon} - 1)}) \frac{1}{1-\varepsilon} \Pr[\mathcal{A}_{\mathcal{O}}^{prsm}(\mathcal{S}') \in \Omega] + 4\delta + \frac{\varepsilon\delta e^2}{1-\delta}$$

For $\varepsilon, \delta \leq 1/2$, $(1 + \frac{e^2}{(1-\delta)^2(\frac{1}{\varepsilon}-1)}) \frac{1}{1-\varepsilon} \leq e^{8e^2\varepsilon + \varepsilon + \varepsilon^2}$, which establishes that **PRSMA** is $(8e^2\varepsilon + \varepsilon + \varepsilon^2, 4\delta + \frac{\varepsilon\delta e^2}{1-\delta})$ differentially private. Setting $\varepsilon = \varepsilon^*, \delta = \delta^*$ completes the proof. □

A.2.4. Proofs from Section 4.5

Theorem 4.5.3. *Let $\varepsilon, \delta \in (0,1)$ and let $\mathcal{A}$ be an (ε, δ) differentially private pERM algorithm for query class $\mathcal{Q}$, with perturbation distribution $\mathcal{D}_{(\varepsilon,\delta)}$. Then Follow the Private Leader instantiated with $\mathcal{A}$ has expected regret bounded by:*

$$R(T) \leq O\left(\varepsilon + \delta + \frac{\mathbb{E}_{Z \sim \mathcal{D}_{\varepsilon,\delta}}[\|Z\|_\infty]}{T}\right)$$

Proof. This theorem is folklore, and this proof is adapted from the lecture notes of Roth and Smith (2017). We introduce some notation. First, write $\ell^t \in \mathbb{R}^{|\mathcal{Q}|}$ to denote the "loss vector" faced by the algorithm at round t, with value $q_i(x^t)$ in coordinate i. Write $\ell^{1:t}$ to denote the summed vector $\ell^{1:t} = \sum_{j=1}^{t} \ell^j$. Write $M : \mathbb{R}^d \to \mathbb{R}^d$ to denote the function such that $M(v)_{i^*} = 1$ where $i^* = \arg\min_i v_i$ and $M(v)_i = 0$ otherwise. In this notation, at each round t, "Follow the Leader" obtains loss $M(\ell^{1:t-1}) \cdot \ell^t$ and "Follow the Private Leader" obtains loss $M(\ell^{1:t-1} + Z^t) \cdot \ell^t$. At the end of play, at time T, the best query q in hindsight obtains cumulative loss $M(\ell^{1:T}) \cdot \ell^{1:T}$.

The proof of this theorem will go through a thought experiment. Consider an imaginary algorithm called "be the leader", which at round t plays according to $M(\ell^{1:t})$. We will first

show that this imaginary algorithm obtains loss that is only lower than that of the best action in hindsight.

Lemma A.2.16 (Kalai and Vempala (2005))**.**

$$\sum_{i=1}^{T} M(\ell^{1:t}) \cdot \ell^t \leq M(\ell^{1:T}) \cdot \ell^{1:T}$$

Proof. This follows by a simple induction on T. For $T = 1$, it holds with equality. Now assume it holds for general T – we show it holds for the next time step:

$$\sum_{i=1}^{T+1} M(\ell^{1:t}) \cdot \ell^t \leq M(\ell^{1:T}) \cdot \ell^{1:T} + M(\ell^{1:T+1}) \cdot \ell^{T+1} \leq$$

$$M(\ell^{1:T+1}) \cdot \ell^{1:T} + M(\ell^{1:T+1}) \cdot \ell^{T+1} = M(\ell^{1:T+1}) \cdot \ell^{1:T+1} \quad \text{(A.12)}$$

□

Recall that private pERM algorithms operate by sampling a perturbation vector $Z^t \sim \mathcal{D}_{\varepsilon,\delta}$ at each round. Next, we show that "be the private leader", which at round t plays according to $M(\ell^{1:t} + Z^t)$, doesn't do much worse.

Lemma A.2.17 (Kalai and Vempala (2005))**.** *For any set of loss vectors $\ell^1, \ldots, \ell^T$ and any set of perturbation vectors $Z^0 \equiv 0, Z^1, \ldots, Z^T$:*

$$\sum_{t=1}^{T} M(\ell^{1:t} + Z^t) \cdot \ell^t \leq M(\ell^{1:T}) \cdot \ell^{1:T} + 2 \sum_{t=1}^{T} \|Z^t - Z^{t-1}\|_\infty$$

Proof. Define $\hat{\ell}^t = \ell^t + Z^t - Z^{t-1}$. Note that $\hat{\ell}^{1:t} = \ell^{1:t} + Z^t$, since the sum telescopes. Thus, we can apply Lemma A.2.16 on the sequence $\hat{\ell}$ to conclude:

$$\begin{aligned}\sum_{t=1}^{T} M(\ell^{1:t} + Z^t) \cdot (\ell^t + Z^t - Z^{t-1}) &\leq M(\ell^{1:T} + Z^T) \cdot (\ell^{1:T} + Z^T) \\ &\leq M(\ell^{1:T}) \cdot (\ell^{1:T} + Z^T) \\ &= M(\ell^{1:T}) \cdot \ell^{1:T} + \sum_{t=1}^{T} M(\ell^{1:T}) \cdot (Z^t - Z^{t-1})\end{aligned}$$

Subtracting from both sides, we have:

$$\sum_{t=1}^{T} M(\ell^{1:t} + Z^t) \cdot \ell^t \leq M(\ell^{1:T}) \cdot \ell^{1:T} + \sum_{t=1}^{T} (M(\ell^{1:T}) - M(\ell^{1:t} + Z^t)) \cdot (Z^t - Z^{t-1}) \leq$$
$$M(\ell^{1:T}) \cdot \ell^{1:T} + \sum_{t=1}^{T} 2\|Z^t - Z^{t-1}\|_\infty \quad \text{(A.13)}$$

□

We will use this lemma and a trick to compute the expected regret of "be the private leader". Since expectations distribute over sums, we have:

$$\mathbb{E}[\sum_{t=1}^{T} M(\ell^{1:t} + Z^t) \cdot \ell^t] = \sum_{t=1}^{T} \mathbb{E}[M(\ell^{1:t} + Z^t) \cdot \ell^t]$$

Hence, the expectation remains unchanged in the thought experiment under which the perturbation is not resampled at every step, and instead $Z^1 = \ldots = Z^t \sim \mathcal{D}_{\varepsilon,\delta}$. Applying Lemma A.2.17 to this version of be the private leader, we obtain:

$$\mathbb{E}[\sum_{t=1}^{T} M(\ell^{1:t} + Z^t) \cdot \ell^t] \leq M(\ell^{1:T}) \cdot \ell^{1:T} + 2\mathbb{E}[\|Z^1\|_\infty]$$

Finally, we use the fact that the algorithm is (ε, δ)-differentially private, and the difference between "follow the private leader" and "be the private leader" amounts to running a

differentially private algorithm on one of two datasets. For any (ε, δ) differentially private algorithm $\mathcal{A} : \mathcal{X}^* \to R$, for any function $f : R \to [0, T]$, and for any pair of neighboring datasets S, S' we have that:

$$\mathbb{E}[f(\mathcal{A}(S))] \leq e^{\varepsilon}\mathbb{E}[f(\mathcal{A}(S'))] + \delta T.$$

We can therefore conclude that for each t:

$$\mathbb{E}[M(\ell^{1:t-1} + Z^t) \cdot \ell^t] \leq e^{\varepsilon}\mathbb{E}[M(\ell^{1:t} + Z^t) \cdot \ell^t] + \delta t$$

Combining this bound with the regret bound we have proven for "be the private leader" yields:

$$\sum_{t=1}^{T} \mathbb{E}[M(\ell^{1:t-1}+Z^t)\cdot\ell^t] \leq e^{\varepsilon}\mathbb{E}[\sum_{t=1}^{T} M(\ell^{1:t}+Z^t)\cdot\ell^t]+\delta T \leq e^{\varepsilon}\left(M(\ell^{1:T})\cdot\ell^{1:T} + 2\mathbb{E}[\|Z^1\|_\infty]\right)+\delta T \leq$$

$$(1+2\varepsilon)\left(M(\ell^{1:T})\cdot\ell^{1:T} + 2\mathbb{E}[\|Z^1\|_\infty]\right) + \delta T \leq M(\ell^{1:T})\cdot\ell^{1:T} + (2+4\varepsilon)\mathbb{E}[\|Z^1\|_\infty] + 2\varepsilon T + \delta T$$

Dividing by T yields the theorem. □

Theorem 4.5.5. *For any d, there is a fixed finite class of statistical queries $\mathcal{Q}$ of size $|\mathcal{Q}| = N = 2^d$ defined over a data universe of size $|\mathcal{X}| = O(N^5 \log^2 N)$ such that for every online learning algorithm with access to a weighted optimization oracle for $\mathcal{Q}$, it cannot guarantee that its expected average regret will be $o(1)$ in total time less than $\Omega(\sqrt{N}/\log^3(N))$.*

Proof. We start by quoting the main ingredient proven in Hazan and Koren (2016) that goes into their lower bound:

Theorem A.2.18 (Hazan and Koren (2016) Theorem 4)**.** *For every $N = 2^d$, and for every randomized algorithm for the players in the game with access to a best-response oracle, there is an $N \times N$ game with payoffs taking values in $\{0, 1/4, 3/4, 1\}$ such that with probability $2/3$ the players*

have not converged to a 1/4*-approximate min-max equilibrium until at least* $\Omega(\sqrt{N}/\log^3(N))$ *time.*

We start by using the Yao min-max principle to reverse the order of quantifiers: Theorem A.2.18 also implies that there is a fixed *distribution* over $N \times N$ games that is hard in expectation for *every* algorithm. However, because we will be interested in the support size of this distribution, we go into a bit more detail in how we apply the min-max principle.

Consider a "meta game" defined by an (infinite) matrix M, with rows i indexed by the $L = 4^{N^2}$ $N \times N$ zero-sum games G_i taking values in $\{0, 1/4, 3/4, 1\}$, and columns indexed by algorithms A_j instantiated with best-response oracles designed to play zero-sum games. $M(i, j)$ will encode the expected running time before algorithm A_j when used to play game G_i converges to a value that is within $1/4$ of the equilibrium value of game G_i. Let the row player (the "lower bound" player) be the maximization player in the zero sum game defined by M, and let the column player (the "algorithm player") be the minimization player. As stated, the entries in M can take unboundedly large values — but observe that there is a simple modification to the game that allows us to upper bound the entries in M by $O(N \log N)$. This is because there exists an algorithm A (the multiplicative weights algorithm — see e.g. Arora et al. (2012)) that can be used to play any $N \times N$ game and converge to a 1/4-approximate equilibrium after time at most $O(N \log N)$ (running in time $O(N)$ per iteration for $O(\log N)$ iterations). It is also possible to check whether a pair of distributions form a 1/4 approximate equilibrium with two calls to a best response oracle. Hence, we can take any algorithm A_i and modify it so that it converges to a $1/4$-approximate equilibrium after at most $O(N \log N)$ time. We simply halt the algorithm after $O(N \log N)$ time if it has not yet converged, and run the multiplicative weights algorithm A. Theorem A.2.18 implies that the value of this modified game M is at least $\Omega(\sqrt{N}/\log^3(N))$.

We now observe that the "meta-game" M has an $O(1)$-approximate max-min strategy for the lower bound player that has support size at most $O(N^4 \log^2 N)$. To see this, consider

the following constructive approach to computing an approximate equilibrium: simulate play of the game in rounds. Let the lower bound player sample a game G^t at each round t using the multiplicative weights distribution over her L actions, and let the algorithm player best respond at each round to the lower bound player's distribution. By construction, the algorithm player has 0 regret, whereas the lower bound player has regret $O(\sqrt{\log L/T} \cdot N \log N)$ after T rounds (since the entries of M are bounded between 0 and $O(N \log N)$) This corresponds to $O(1)$ regret after $T = O(\log L \cdot N^2 \log^2 N) = O(N^4 \log^2 N)$ many rounds. By Theorem 5.4.9, the empirical distribution over these $O(N^4 \log^2 N)$ many games G^t forms an $O(1)$-approximate max-min strategy. Thus we have proven:

Corollary A.2.19. *For every $N = 2^d$, there is a fixed set $H \subseteq \{0, 1/4, 3/4, 1\}^{N \times N}$ of $N \times N$ games, of size $|H| = O(N^4 \log^2 N)$ such that for every randomized algorithm A for players in a game with access to a best response oracle, there is a game $G \in H$ such that with probability $2/3$, the players have not converged to a $1/4$-approximate min-max equilibrium until at least $\Omega(\sqrt{N}/\log^3(N))$ time.*

Now consider the $N \times O(N^5 \log^2 N)$ matrix R that results from stacking the matrices in H. Identify the N rows with a query class $\mathcal{Q}$ of size N, indexed by functions $q \in \mathcal{Q}$. Identify the columns with a data universe $\mathcal{X}$ of size $|\mathcal{X}| = O(N^5 \log^2 N)$ indexed by $x \in \mathcal{X}$, and define the queries such that $q(x) = R(q, x)$ for each $q \in \mathcal{Q}, x \in \mathcal{X}$. Observe that any no-regret algorithm with action set $\mathcal{Q}$ that can obtain $o(1)$ regret against an adversary who is constrained to play loss vectors in $\mathcal{X}$ can be used to compute an $o(1)$ approximate equilibrium strategy for any game in H (together with a single call to a best-response oracle per round by his oppoinent, by Theorem 5.4.9). Thus we can conclude that no algorithm can guarantee to get $o(1)$ regret over $\mathcal{Q}$ until at least $\Omega(\sqrt{N}/\log^3(N))$ time. □

Theorem 4.5.6. *Any oracle efficient (i.e. running in time poly$(n, \log|\mathcal{Q}|)$) (ε, δ)-differentially private pERM algorithm instantiated with a weighted optimization oracle for the query class $\mathcal{Q}$ defined in Theorem 4.5.5, with perturbation distribution $\mathcal{D}_{(\varepsilon,\delta)}$ must be such that for every*

$(\varepsilon + \delta) = o(1)$*:*

$$\mathbb{E}_{Z \sim \mathcal{D}_{(\varepsilon,\delta)}}[\|Z\|_\infty] \geq \Omega(|\mathcal{Q}|^c)$$

for some constant $c > 0$.

Proof. For any such pERM algorithm $\mathcal{A}$, Let $B = \mathbb{E}_{Z \sim \mathcal{D}_{(\varepsilon,\delta)}}[\|Z\|_\infty]$. We know from Theorem 4.5.3 that follow the private leader instantiated with $\mathcal{A}$ obtains regret $o(1)$ whenever $T = \omega(B)$, for any $\varepsilon + \delta = o(1)$. Since $\mathcal{A}$ is oracle efficient (i.e. runs in time $\text{poly}(t, \log|\mathcal{Q}|)$), the total running time needed to obtain diminishing regret is $\sum_{t=1}^{T} \text{poly}(t, \log|\mathcal{Q}|) = \text{poly}(B, \log|\mathcal{Q}|)$. We know from Theorem 4.5.5 that to guarantee diminishing regret over $\mathcal{Q}$, the total running time must be at least $\Omega(\sqrt{|\mathcal{Q}|}/\log^3(|\mathcal{Q}|))$. Thus we must have that $B = \text{poly}(|\mathcal{Q}|)$ — i.e. $B = \Omega(|Q|^c)$ for some $c > 0$. □

A.3. Details from Chapter 5

A.3.1. Chernoff-Hoeffding Bound

We use the following concentration inequality.

Theorem A.3.1 (Real-vaued Additive Chernoff-Hoeffding Bound)**.** *Let* $X_1, X_2, \ldots, X_m$ *be i.i.d. random variables with* $\mathbb{E}[X_i] = \mu$ *and* $a \leq X_i \leq b$ *for all* i*. Then for every* $\alpha > 0$*,*

$$\Pr\left[\left|\frac{\sum_i X_i}{m} - \mu\right| \geq \alpha\right] \leq 2\exp\left(\frac{-2\alpha^2 m}{(b-a)^2}\right)$$

A.3.2. Generalization Bounds

Proof of Theorems 5.2.8 and 5.2.9. We give a proof of Theorem 5.2.8. The proof of Theorem 5.2.9 is identical, as false positive rates are just positive classification rates on the subset of the data for which $y = 0$.

Given a set of classifiers $\mathcal{H}$ and protected groups $\mathcal{G}$, define the following function class:

$$\mathcal{F}_{\mathcal{H},\mathcal{G}} = \{f_{h,g}(x) \doteq h(x) \wedge g(x) : h \in \mathcal{H}, g \in \mathcal{G}\}$$

We can relate the VC-dimension of $\mathcal{F}_{\mathcal{H},\mathcal{G}}$ to the VC-dimension of $\mathcal{H}$ and $\mathcal{G}$:

Claim A.3.2.

$$\text{VCDIM}(\mathcal{F}_{\mathcal{H},\mathcal{G}}) \leq \tilde{O}(\text{VCDIM}(\mathcal{H}) + \text{VCDIM}(\mathcal{G}))$$

Proof. Let S be a set of size m shattered by $\mathcal{F}_{\mathcal{H},\mathcal{G}}$. Let $\pi_{\mathcal{F}_{\mathcal{H},\mathcal{G}}}(S)$ be the number of labelings of S realized by elements of $\mathcal{F}_{\mathcal{H},\mathcal{G}}$. By the definition of shattering, $\pi_{\mathcal{F}_{\mathcal{H},\mathcal{G}}}(S) = 2^m$. Now for each labeling of S by an element in $\mathcal{F}_{\mathcal{H},\mathcal{G}}$, it is realized as $(f \wedge g)(S)$ for some $f \in \mathcal{F}, g \in \mathcal{G}$. But $(f \wedge g)(S) = f(S) \wedge g(S)$, and so it can be realized as the conjunction of a labeling of S by an element of $\mathcal{F}$ and an element of $\mathcal{G}$. But since there are $\pi_{\mathcal{F}}(S)\pi_{\mathcal{G}}(S)$ such pairs of labelings, this immediately implies that $\pi_{\mathcal{F}_{\mathcal{H},\mathcal{G}}}(S) \leq \pi_{\mathcal{F}}(S)\pi_{\mathcal{G}}(S)$. Now by the Sauer-Shelah Lemma (see e.g. Kearns and Vazirani (1994b)), $\pi_{\mathcal{F}}(S) = O(m^{\text{VCDIM}(\mathcal{H})}), \pi_{\mathcal{G}}(S) = O(m^{\text{VCDIM}(\mathcal{G})})$. Thus $\pi_{\mathcal{F}_{\mathcal{H},\mathcal{G}}}(S) = 2^m \leq O(m^{\text{VCDIM}(\mathcal{H})+\text{VCDIM}(\mathcal{G})})$, which implies that $m = \tilde{O}(\text{VCDIM}(\mathcal{H})+\text{VCDIM}(\mathcal{G}))$, as desired.

□

This bound, together with a standard VC-Dimension based uniform convergence theorem (see e.g. Kearns and Vazirani (1994b)) implies that with probability $1-\delta$, for every $f_{h,g} \in \mathcal{F}_{\mathcal{H},\mathcal{G}}$:

$$\left|\mathbb{E}_{(X,y)\sim\mathcal{P}}[f_{h,g}(X)] - \mathbb{E}_{(X,y)\sim\mathcal{P}_S}[f_{h,g}(X)]\right| \leq \tilde{O}\left(\sqrt{\frac{(\text{VCDIM}(\mathcal{H}) + \text{VCDIM}(\mathcal{G}))\log m + \log(1/\delta)}{m}}\right)$$

Note that the left hand side of the above inequality can be written as:

$$\left|\Pr_{(X,y)\sim\mathcal{P}}[h(X) = 1 | g(x) = 1] \cdot \Pr_{(X,y)\sim\mathcal{P}}[g(x) = 1] - \Pr_{(X,y)\sim\mathcal{P}_S}[h(X) = 1 | g(x) = 1] \cdot \Pr_{(X,y)\sim\mathcal{P}_S}[g(x) = 1]\right|$$

This completes our proof. □

A.3.3. Missing Proofs in Section 5.4

Theorem 5.4.5. *Let $(\hat{D}, \hat{\lambda})$ be a ν-approximate minmax solution to the Λ-bounded Lagrangian problem in the sense that*

$$\mathcal{L}(\hat{D}, \hat{\lambda}) \leq \min_{D \in \Delta_{\mathcal{H}(S)}} \mathcal{L}(D, \hat{\lambda}) + \nu \quad \textit{and,} \quad \mathcal{L}(\hat{D}, \hat{\lambda}) \geq \max_{\lambda \in \Lambda} \mathcal{L}(\hat{D}, \lambda) - \nu.$$

Then $err(\hat{D}, \mathcal{P}) \leq \mathrm{OPT} + 2\nu$ and for any $g \in \mathcal{G}(S)$,

$$\alpha_{FP}(g, \mathcal{P})\, \beta_{FP}(g, \hat{D}, \mathcal{P}) \leq \gamma + \frac{1 + 2\nu}{C}.$$

Proof of Theorem 5.4.5. Let D^* be the optimal feasible solution for our constrained optimization problem. Since D^* is feasible, we know that $\mathcal{L}(D^*, \hat{\lambda}) \leq err(D^*, \mathcal{P})$.

We will first focus on the case where $\hat{D}$ is not a feasible solution, that is

$$\max_{(g, \bullet) \in \mathcal{G}(S) \times \{\pm\}} \Phi_{\bullet}(\hat{D}, g) > 0$$

Let $(\hat{g}, \hat{\bullet}) \in \mathrm{argmax}_{(g, \bullet)} \Phi_{\bullet}(\hat{D}, g)$ and let $\lambda' \in \Lambda$ be a vector with $(\lambda')^{\hat{\bullet}}_{\hat{g}} = C$ and all other coordinates zero. By Lemma 5.4.8, we know that $\lambda' \in \mathrm{argmax}_{\lambda \in \Lambda} \mathcal{L}(\hat{D}, \lambda)$. By the definition of a ν-approximate minmax solution, we know that $\mathcal{L}(\hat{D}, \hat{\lambda}) \geq \mathcal{L}(\hat{D}, \lambda') - \nu$. This implies that

$$\mathcal{L}(\hat{D}, \hat{\lambda}) \geq err(\hat{D}, \mathcal{P}) + C\, \Phi_{\hat{\bullet}}(\hat{D}, \hat{g}) - \nu \tag{A.14}$$

Note that $\mathcal{L}(D^*, \hat{\lambda}) \leq err(D^*, \mathcal{P})$, and so

$$\mathcal{L}(\hat{D}, \hat{\lambda}) \leq \min_{D \in \Delta_{\mathcal{H}(S)}} \mathcal{L}(D, \hat{\lambda}) + \nu \leq \mathcal{L}(D^*, \hat{\lambda}) + \nu \tag{A.15}$$

Combining Equations (A.14) and (A.15), we get

$$err(\hat{D},\mathcal{P}) + C\,\Phi_{\hat{\bullet}}(\hat{D},\hat{g}) \leq \mathcal{L}(\hat{D},\hat{\lambda}) + \nu \leq \mathcal{L}(D^*,\hat{\lambda}) + 2\nu \leq err(D^*,\mathcal{P}) + 2\nu$$

Note that $C\,\Phi_{\hat{\bullet}}(\hat{D},\hat{g}) \geq 0$, so we must have $err(\hat{D},\mathcal{P}) \leq err(D^*,\mathcal{P}) + 2\nu = \text{OPT} + 2\nu$. Furthermore, since $err(\hat{D},\mathcal{P}), err(D^*,\mathcal{P}) \in [0,1]$, we know

$$C\,\Phi_{\hat{\bullet}}(\hat{D},\hat{g}) \leq 1 + 2\nu,$$

which implies that maximum constraint violation satisfies $\Phi_{\hat{\bullet}}(\hat{D},\hat{g}) \leq (1+2\nu)/C$. By applying Claim 5.4.4, we get

$$\alpha_{FP}(g,\mathcal{P})\;\beta_{FP}(g,\hat{D},\mathcal{P}) \leq \gamma + \frac{1+2\nu}{C}.$$

Now let us consider the case in which $\hat{D}$ is a feasible solution for the optimization problem. Then it follows that there is no constraint violation by $\hat{D}$ and $\max_\lambda \mathcal{L}(\hat{D},\lambda) = err(\hat{D},\mathcal{P})$, and so

$$err(\hat{D},\mathcal{P}) = \max_{\lambda} \mathcal{L}(\hat{D},\lambda) \leq \mathcal{L}(\hat{D},\hat{\lambda}) + \nu \leq \min_{D} \mathcal{L}(D,\hat{\lambda}) + 2\nu \leq \mathcal{L}(D^*,\hat{\lambda}) + 2\nu \leq err(D^*,\mathcal{P}) + 2\nu$$

Therefore, the stated bounds hold for both cases. □

Lemma 5.4.8. *Fix any $\overline{D} \in \Delta_{\mathcal{H}(S)}$ such that that $\max_{g\in\mathcal{G}(S)}\{\Phi_+(\overline{D},g),\Phi_-(\overline{D},g)\} > 0$. Let $\lambda' \in \Lambda$ be vector with one non-zero coordinate $(\lambda')^{\bullet'}_{g'} = C$, where*

$$(g',\bullet') = \operatorname*{argmax}_{(g,\bullet)\in\mathcal{G}(S)\times\{\pm\}} \{\Phi_\bullet(\overline{D},g)\}$$

Then $\mathcal{L}(\overline{D},\lambda') \geq \max_{\lambda\in\Lambda} \mathcal{L}(\overline{D},\lambda)$.

Proof of Lemma 5.4.8. Observe:

$$\begin{aligned}\operatorname*{argmax}_{\lambda\in\Lambda}\mathcal{L}(\overline{D},\lambda) &= \operatorname*{argmax}_{\lambda\in\Lambda}\mathop{\mathbb{E}}_{h\sim\overline{D}}[err(h,\mathcal{P})] + \sum_{g\in\mathcal{G}(S)}\left(\lambda_g^+\Phi_+(\overline{D},g)+\lambda_g^-\Phi_-(\overline{D},g)\right)\\ &= \operatorname*{argmax}_{\lambda\in\Lambda}\sum_{g\in\mathcal{G}}\left(\lambda_g^+\Phi_+(\overline{D},g)+\lambda_g^-\Phi_-(\overline{D},g)\right)\end{aligned}$$

Note that this is a linear optimization problem over the non-negative orthant of a scaling of the ℓ_1 ball, and so has a solution at a vertex, which corresponds to a single group $g \in \mathcal{G}(S)$. Thus, there is always a best response λ' that puts all the weight C on the coordinate $(\lambda')^\bullet_g$ that maximizes $\Phi_\bullet(\overline{D},g)$. □

Lemma 5.4.10. *Let T be the time horizon for the no-regret dynamics. Let $D^1,\ldots,D^T$ be the sequence of distributions maintained by the Learner's FTPL algorithm with $\eta = \frac{n}{(1+C)}\sqrt{\frac{1}{\sqrt{n}T}}$, and $\lambda^1,\ldots,\lambda^T$ be the sequence of plays by the Auditor. Then*

$$\sum_{t=1}^{T}\mathop{\mathbb{E}}_{h\sim D^t}\left[U(h,\lambda^t)\right] - \min_{h\in\mathcal{H}(S)}\sum_{t=1}^{T}U(h,\lambda^t) \le 2n^{1/4}(1+C)\sqrt{T}$$

Proof of Lemma 5.4.10. To instantiate the regret bound in Theorem 5.2.6, we just need to provide a bound on the maximum absoluate value over the coordinates of the loss vector (the quantity M in Theorem 5.2.6). For any $\lambda \in \Lambda$, the absolute value of the i-th coordinate of $\mathrm{LC}(\lambda)$ is bounded by:

$$\begin{aligned}&\frac{1}{n} + \left|\frac{1}{n}\sum_{g\in\mathcal{G}(S)}(\lambda_g^+ - \lambda_g^-)\left(\Pr[g(x)=1 \mid y=0]-1\right)\mathbf{1}[g(x_i)=1]\right|\\ \le&\frac{1}{n} + \frac{1}{n}\left(\sum_{g\in\mathcal{G}(S)}\left|\lambda_g^+ - \lambda_g^-\right|\right)\max_{g\in\mathcal{G}(S)}\left(\Pr[g(x)=1 \mid y=0]\mathbf{1}g(x_i)=1\right)\\ \le&\frac{1}{n} + \frac{1}{n}\left(\sum_{g\in\mathcal{G}(S)}\left|\lambda_g^+\right| + \left|\lambda_g^-\right|\right) \le \frac{1+C}{n}\end{aligned}$$

Also note that the dimension of the optimization is the size of the dataset n. This means if we set $\eta = \frac{n}{(1+C)}\sqrt{\frac{1}{\sqrt{n}T}}$, the regret of the learner will then be bounded by $2n^{1/4}(1+C)\sqrt{T}$. □

Lemma 5.4.11. *Fix any $\xi, \delta \in (0,1)$ and any distribution D over $\mathcal{H}(S)$. Let $h^1, \ldots, h^m$ be m i.i.d. draws from p, and $\hat{D}$ be the empirical distribution over the realized sample. Then with probability at least $1-\delta$ over the random draws of h^j's, the following holds,*

$$\max_{\lambda \in \Lambda} \left| \mathop{\mathbb{E}}_{h \sim \hat{D}} [U(h,\lambda)] - \mathop{\mathbb{E}}_{h \sim D} [U(h,\lambda)] \right| \leq \xi,$$

as long as $m \geq c_0 \frac{C^2(\ln(1/\delta)+d_2 \ln(n))}{\xi^2}$ for some absolute constant c_0 and $d_2 = \text{VCDIM}(\mathcal{G})$.

Proof of Lemma 5.4.11. Recall that for any distribution D' over $\mathcal{H}(S)$ the expected payoff function is defined as

$$\begin{aligned} \mathop{\mathbb{E}}_{h \sim \hat{D}} [U(h,\lambda)] - \mathop{\mathbb{E}}_{h \sim D} [U(h,\lambda)] = & \mathop{\mathbb{E}}_{h \sim \hat{D}} [err(h,\mathcal{P})] + \mathop{\mathbb{E}}_{h \sim \hat{D}} \left[\sum_{g \in \mathcal{G}(S)} \left(\lambda_g^+ \Phi_+(h,g) + \lambda_g^- \Phi_-(h,g) \right) \right] \\ & - \mathop{\mathbb{E}}_{h \sim D} [err(h,\mathcal{P})] + \mathop{\mathbb{E}}_{h \sim D} \left[\sum_{g \in \mathcal{G}(S)} \left(\lambda_g^+ \Phi_+(h,g) + \lambda_g^- \Phi_-(h,g) \right) \right] \end{aligned}$$

By the triangle inequality, it suffices to show that with probability $(1-\delta)$, $A = |\mathbb{E}_{h \sim D} [err(h,\mathcal{P})] - \mathbb{E}_{h \sim \hat{D}} [err(h,\mathcal{P})]| \leq \xi/2$ and for all $\lambda \in \Lambda$ and $g \in \mathcal{G}(S)$,

$$B = \left| \mathop{\mathbb{E}}_{h \sim \hat{D}} \left[\sum_{g \in \mathcal{G}(S)} \left(\lambda_g^+ \Phi_+(h,g) + \lambda_g^- \Phi_-(h,g) \right) \right] - \mathop{\mathbb{E}}_{h \sim D} \left[\sum_{g \in \mathcal{G}(S)} \left(\lambda_g^+ \Phi_+(h,g) + \lambda_g^- \Phi_-(h,g) \right) \right] \right| \leq \xi/2$$

The first part follows directly from a simple application of the Chernoff-Hoeffding bound (Theorem A.3.1): with probability $(1-\delta/2)$, $A \leq \xi/2$, as long as $m \geq 2\ln(4/\delta)/\xi^2$.

To bound the second part, we first note that by Hölder's inequality, we have

$$B \leq \|\lambda\|_1 \max_{(g,\bullet) \in \mathcal{G}(S) \times \{\pm\}} |\Phi_\bullet(D,g) - \Phi_\bullet(\hat{D},g)|$$

Since for all $\lambda \in \Lambda$ we have $\|\lambda\|_1 \leq C$, it suffices to show that with probability $1 - \delta/2$, $|\Phi_\bullet(D, g) - \Phi_\bullet(\hat{D}, g)| \leq \xi/(2C)$ holds for all $\bullet \in \{-, +\}$ and $g \in \mathcal{G}(S)$. Note that

$$|\Phi_\bullet(D, g) - \Phi_\bullet(\hat{D}, g)| = \left| \left(\mathop{\mathbb{E}}_{h \sim D} [\mathrm{FP}(h)] - \mathop{\mathbb{E}}_{h \sim \hat{D}} [\mathrm{FP}(h)] \right) \Pr[y = 0, g(x) = 1] \right.$$
$$\left. + \left(\mathop{\mathbb{E}}_{h \sim D} [\Pr[h(X) = 1, y = 0, g(x) = 1]] - \mathop{\mathbb{E}}_{h \sim \hat{D}} [\Pr[h(X) = 1, y = 0, g(x) = 1]] \right) \right|$$

We can rewrite the absolute value of first term:

$$\begin{aligned}
&\left| \left(\mathop{\mathbb{E}}_{h \sim D} [\mathrm{FP}(h)] - \mathop{\mathbb{E}}_{h \sim \hat{D}} [\mathrm{FP}(h)] \right) \Pr[y = 0, g(x) = 1] \right| \\
&= \left| \left(\mathop{\mathbb{E}}_{h \sim D} [\Pr[h(X) = 1 \mid y = 0]] - \mathop{\mathbb{E}}_{h \sim \hat{D}} [\Pr[h(X) = 1 \mid y = 0]] \right) \Pr[g(x) = 1 \mid y = 0] \right| \\
&\leq \left| \left(\mathop{\mathbb{E}}_{h \sim D} [\Pr[h(X) = 1, y = 0]] - \mathop{\mathbb{E}}_{h \sim \hat{D}} [\Pr[h(X) = 1, y = 0]] \right) \right|
\end{aligned}$$

where the last inequality follows from $\Pr[g(x) = 1 \mid y = 0] \leq 1$.

Note that $\mathbb{E}_{h \sim \hat{D}} [\Pr[h(X) = 1, y = 0, g(x) = 1]] = \frac{1}{m} \sum_{j=1}^{m} \Pr[h^j(X) = 1, y = 0, g(x) = 1]$, which is an average of m i.i.d. random variables with expectation $\mathbb{E}_{h \sim D} [\Pr[h(X) = 1, y = 0, g(x) = 1]]$. By the Chernoff-Hoeffding bound (Theorem A.3.1), we have

$$\Pr \left[\left| \mathop{\mathbb{E}}_{h \sim D} [\Pr[h(X) = 1, y = 0]] - \mathop{\mathbb{E}}_{h \sim \hat{D}} [\Pr[h(X) = 1, y = 0]] \right| > \frac{\xi}{4C} \right] \leq 2 \exp \left(- \frac{\xi^2 m}{8C^2} \right) \tag{A.16}$$

In the following, we will let $\delta_0 = 2 \exp \left(- \frac{\xi^2 m}{8C^2} \right)$. Similarly, we also have for each $g \in \mathcal{G}(S)$,

$$\Pr \left[\left| \mathop{\mathbb{E}}_{h \sim D} [\Pr[h(X) = 1, y = 0, g(x) = 1]] - \mathop{\mathbb{E}}_{h \sim \hat{D}} [\Pr[h(X) = 1, y = 0, g(x) = 1]] \right| > \frac{\xi}{4C} \right] \leq \delta_0 \tag{A.17}$$

By taking the union bound over (A.16) and (A.17) over all choices of $g \in \mathcal{G}(S)$, we have with probability at least $(1 - \delta_0(1 + |\mathcal{G}(S)|))$,

$$\left| \mathop{\mathbb{E}}_{h \sim D} [\Pr[h(X) = 1, y = 0]] - \mathop{\mathbb{E}}_{h \sim \hat{D}} [\Pr[h(X) = 1, y = 0]] \right| \leq \frac{\xi}{4C} \tag{A.18}$$

and,

$$\left|\mathop{\mathbb{E}}_{h\sim D}[\Pr[h(X)=1,y=0,g(x)=1]]-\mathop{\mathbb{E}}_{h\sim \hat{D}}[\Pr[h(X)=1,y=0,g(x)=1]]\right|\leq \frac{\xi}{4C} \quad \text{for all } g\in\mathcal{G}(S). \tag{A.19}$$

Note that by Sauer's lemma (Lemma 5.4.3), $|\mathcal{G}(S)|\leq O\left(n^{d_2}\right)$. Thus, there exists an absolute constant c_0 such that $m\geq c_0\frac{C^2(\ln(1/\delta)+d_2\ln(n))}{\xi^2}$ implies that failure probability above $\delta_0(1+|\mathcal{G}(S)|)\leq\delta/2$. We will assume m satisifies such a bound, and so the events of (A.18) and (A.19) hold with probaility at least $(1-\delta/2)$. Then by the triangle inequality we have for all $(g,\bullet)\in\mathcal{G}(S)\times\{\pm\}$, $|\Phi_\bullet(D,g)-\Phi_\bullet(\hat{D},g)|\leq\xi/(2C)$, which implies that $B\leq\xi/2$. This completes the proof. □

Claim A.3.3. *Suppose there are two distributions D and $\hat{D}$ over $\mathcal{H}(S)$ such that*

$$\max_{\lambda\in\Lambda}\left|\mathop{\mathbb{E}}_{h\sim\hat{D}}[U(h,\lambda)]-\mathop{\mathbb{E}}_{h\sim D}[U(h,\lambda)]\right|\leq\xi.$$

Let

$$\hat{\lambda}\in\operatorname*{argmax}_{\lambda'\in\Lambda}\mathop{\mathbb{E}}_{h\sim\hat{D}}[U(h,\lambda')]$$

Then

$$\max_{\lambda}\mathop{\mathbb{E}}_{h\sim D}[U(h,\lambda)]-\xi\leq\mathop{\mathbb{E}}_{h\sim D}\left[U(h,\hat{\lambda})\right],$$

Lemma 5.4.12. *Let T be the time horizon for the no-regret dynamics. Let $D^1,\ldots,D^T$ be the sequence of distributions maintained by the Learner's FTPL algorithm. For each D^t, let $\hat{D}^t$ be the empirical distribution over m i.i.d. draws from D^t. Let $\lambda^1,\ldots,\lambda^T$ be the Auditor's best responses against $\hat{D}^1,\ldots,\hat{D}^T$. Then with probability $1-\delta$,*

$$\max_{\lambda\in\Lambda}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{h\sim D^t}[U(h,\lambda)]-\sum_{t=1}^{T}\mathop{\mathbb{E}}_{h\sim D^t}\left[U(h,\lambda^t)\right]\leq T\sqrt{\frac{c_0C^2(\ln(T/\delta)+d_2\ln(n))}{m}}$$

for some absolute constant c_0 and $d_2=\mathrm{VCDIM}(\mathcal{G})$.

Proof. Let γ_A^t be defined as

$$\gamma_A^t = \max_{\lambda \in \Lambda} \left| \mathop{\mathbb{E}}_{h \sim \hat{D}^t} [U(h, \lambda)] - \mathop{\mathbb{E}}_{h \sim D^t} [U(h, \lambda)] \right|$$

By instantiating Lemma 5.4.11 and applying union bound across all T steps, we know with probability at least $1 - \delta$, the following holds for all $t \in [T]$:

$$\gamma_A^t \leq \sqrt{\frac{c_0 C^2 (\ln(T/\delta) + d_2 \ln(n))}{m}}$$

where c_0 is the absolute constant in Lemma 5.4.11 and $d_2 = \text{VCDIM}(\mathcal{G})$.

Note that by Claim A.3.3, the Auditor is performing a γ_A^t-approximate best response at each round t. Then we can bound the Auditor's regret as follows:

$$\gamma_A = \frac{1}{T} \left[\max_{\lambda \in \Lambda} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{h \sim D^t} [U(h, \lambda)] - \sum_{t=1}^{T} \mathop{\mathbb{E}}_{h \sim D^t} \left[U(h, \lambda^t) \right] \right] \leq$$
$$\frac{1}{T} \sum_{t=1}^{T} \left(\max_{\lambda \in \Lambda} \mathop{\mathbb{E}}_{h \sim D^t} [U(h, \lambda)] - \mathop{\mathbb{E}}_{h \sim D^t} \left[U(h, \lambda^t) \right] \right) \leq \max_T \gamma_A^t$$

It follows that with probability $1 - \delta$, we have

$$\gamma_A \leq \sqrt{\frac{c_0 C^2 (\ln(T/\delta) + d_2 \ln(n))}{m}}$$

which completes the proof. □

A.4. Details from Chapter 6

Lemma A.4.1. *For fixed D and α, the best response optimization for the dual player is separable, i.e.*

$$\operatorname*{argmax}_{\lambda \in \Lambda, \tau \in \mathrm{T}} \mathcal{L}(D, \alpha, \lambda, \tau) = \operatorname*{argmax}_{\lambda \in \Lambda} \mathcal{L}_{D,\alpha}^{\psi_1}(\lambda) \times \operatorname*{argmax}_{\tau \in \mathrm{T}} \mathcal{L}_{D,\alpha}^{\psi_2}(\tau),$$

where

$$\mathcal{L}^{\psi_1}_{D,\alpha}(\lambda) = \sum_{(i,j)\in[n]^2} \lambda_{ij}\left(\mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] - \alpha_{ij} - \gamma\right)$$

and

$$\mathcal{L}^{\psi_2}_{D,\alpha}(\tau) = \tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta\right).$$

Proof.

$$\begin{aligned}
&\operatorname*{argmax}_{\lambda\in\Lambda,\tau\in\mathrm{T}} \mathcal{L}(D,\alpha,\lambda,\tau) \\
&= \operatorname*{argmax}_{\lambda\in\Lambda,\tau\in\mathrm{T}} \mathbb{E}_{h\sim D}\left[err(h,S)\right] + \sum_{(i,j)\in[n]^2} \lambda_{ij}\left(\mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] - \alpha_{ij} - \gamma\right) + \tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta\right) \\
&= \operatorname*{argmax}_{\lambda\in\Lambda} \sum_{(i,j)\in[n]^2} \lambda_{ij}\left(\mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] - \alpha_{ij} - \gamma\right) \times \operatorname*{argmax}_{\tau\in\mathrm{T}} \tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta\right) \\
&= \operatorname*{argmax}_{\lambda\in\Lambda} \mathcal{L}^{\psi_1}_{D,\alpha}(\lambda) \times \operatorname*{argmax}_{\tau\in\mathrm{T}} \mathcal{L}^{\psi_2}_{D,\alpha}(\tau)
\end{aligned}$$

□

Algorithm 12 Best Response, $BEST_\psi(D,\alpha)$, for the dual player

Input: training examples $S = \{x_i, y_i\}_{i=1}^n$, $D \in \Delta(H)$, $\alpha \in [0,1]^{n^2}$
$\lambda = 0 \in \mathbb{R}^{n^2}$
$(i^*, j^*) = \operatorname{argmax}_{(i,j)\in[n]^2} \mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] - \alpha_{ij} - \gamma$
if $\mathbb{E}_{h\sim D}\left[h(x_{i^*}) - h(x_{j^*})\right] - \alpha_{i^*j^*} - \gamma \le 0$ **then**
    $\lambda_{i^*j^*} = C_\lambda$
set $\tau = \begin{cases} 0 & \frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta \le 0 \\ C_\tau & o.w. \end{cases}$
Output: λ, τ

Lemma A.4.2. *For fixed D and α, the output λ from $BEST_\psi(D,\alpha)$ minimizes $\mathcal{L}^{\psi_1}_{D,\alpha}$*

Proof. Because $\mathcal{L}^{\psi_1}_{D,\alpha}$ is linear in terms of λ and the feasible region is the non-negative orthant bounded by 1-norm, the optimal solution must include putting all the weight to the pair (i,j) where $\mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j) - \alpha_{ij}\right]$ is maximized. □

Lemma A.4.3. *For fixed D and α, the output τ from $BEST_{\psi}(D, \alpha)$ minimizes $\mathcal{L}_{D,\alpha}^{\psi_2}$*

Proof. Because $\mathcal{L}_{D,\alpha}^{\psi_2}$ is linear in terms of τ, the optimal solution is trivially to set τ at either C_τ or 0 depending on the sign. □

Bibliography

Sam Levin. A beauty contest was judged by ai and the robots didn't like dark skin, 2016. URL https://www.theguardian.com/technology/2016/sep/08/artificial-intelligence-beauty-contest-doesnt-like-black-people.

Stephen M. Dane. Fair housing policy under the trump adminstration, 2019. URL https://www.americanbar.org/groups/crsj/publications/human_rights_magazine_home/economic-justice/fair-housing-policy-under-the-trump-administration/.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. pages 111–125, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3168-7. doi: http://dx.doi.org/10.1109/SP.2008.33.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *Propublica*, 2016. URL https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Cynthia Rudin. Predictive policing using machine learning to detect patterns of crime. *Wired Magazine*, August 2013. URL http://www.wired.com/insights/2013/08/predictive-policing-using-machine-learning-to-detect-\patterns-of-crime/. Retrieved 4/28/2016.

Claire Cain Miller. Algorithms and bias: Q. and a. with cynthia dwork. New York Times, August 10 2015a.

Clair C Miller. Can an algorithm hire better than a human? *The New York Times*, June 25 2015b. URL http://www.nytimes.com/2015/06/26/upshot/can-an-algorithm-hire-better-than-a-human.html. Retrieved 4/28/2016.

John Davis and Osonde Osoba. Privacy preservation in the age of big data. *RAND: Working Papers*, 2016. URL https://www.rand.org/pubs/working_papers/WR1161.html.

Josep Domingo-Ferrer and Vicenç Torra. A critique of k-anonymity and some of its enhancements. pages 990–993, 03 2008. doi: 10.1109/ARES.2008.97.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. 2006a.

Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, James Honaker, Kobbi Nissim, David O'Brien, Thomas Steinke, and Salil Vadhan. Differential privacy: A primer for a non-technical audience. *SSRN Electronic Journal*, 01 2018. doi: 10.2139/ssrn.3338027.

Cynthia Dwork and Guy N. Rothblum. Concentrated differential privacy. *CoRR*, abs/1603.01887, 2016. URL http://arxiv.org/abs/1603.01887.

Ilya Mironov. Renyi differential privacy. *CoRR*, abs/1702.07476, 2017. URL http://arxiv.org/abs/1702.07476.

Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy. *CoRR*, abs/1905.02383, 2019. URL http://arxiv.org/abs/1905.02383.

Justin Hsu, Zhiyi Huang, Aaron Roth, Tim Roughgarden, and Zhiwei Steven Wu. Private matchings and allocations. *CoRR*, abs/1311.2828, 2013a. URL http://arxiv.org/abs/1311.2828.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. *CoRR*, abs/1506.02629, 2015a. URL http://arxiv.org/abs/1506.02629.

Seth Neel and Aaron Roth. Mitigating bias in adaptive data gathering via differential privacy, 2018.

Christopher Jung, Katrina Ligett, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Moshe Shenfeld. A new analysis of differential privacy's generalization guarantees, 2019.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011. URL http://dl.acm.org/citation.cfm?id=2021036.

Daniel Kifer, Adam D. Smith, and Abhradeep Thakurta. Private convex optimization for empirical risk minimization with applications to high-dimensional regression. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pages 25.1–25.40, 2012. URL http://www.jmlr.org/proceedings/papers/v23/kifer12/kifer12.pdf.

Benjamin I. P. Rubinstein, Peter L. Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for SVM learning. *CoRR*, abs/0911.5708, 2009. URL http://arxiv.org/abs/0911.5708.

Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 289–296, 2008. URL http://papers.nips.cc/paper/3486-privacy-preserving-logistic-regression.

Adam Smith, Jalaj Upadhyay, and Abhradeep Thakurta. Is interaction necessary for distributed private learning? *IEEE Symposium on Security and Privacy*, 2017.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007.

Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization, revisited. *CoRR*, abs/1405.7085, 2014. URL http://arxiv.org/abs/1405.7085.

Oliver Williams and Frank McSherry. Probabilistic inference and differential privacy. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural*

Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada., pages 2451–2459, 2010. URL http://papers.nips.cc/paper/3897-probabilistic-inference-and-differential-privacy.

Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pages 24.1–24.34, 2012. URL http://www.jmlr.org/proceedings/papers/v23/jain12/jain12.pdf.

John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In *51st Annual Allerton Conference on Communication, Control, and Computing, Allerton 2013, Allerton Park & Retreat Center, Monticello, IL, USA, October 2-4, 2013*, page 1592, 2013. doi: 10.1109/Allerton.2013.6736718. URL http://dx.doi.org/10.1109/Allerton.2013.6736718.

Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. In *IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013, Austin, TX, USA, December 3-5, 2013*, pages 245–248, 2013. doi: 10.1109/GlobalSIP.2013.6736861. URL http://dx.doi.org/10.1109/GlobalSIP.2013.6736861.

Anna Maria Barry-Jester, Ben Casselman, and Dana Goldstein. The new science of sentencing. *The Marshall Project*, August 8 2015. URL https://www.themarshallproject.org/2015/08/04/the-new-science-of-sentencing. Retrieved 4/28/2016.

James Rufus Koren. What does that web search say about your credit? *Los Angeles Times*, July 16 2016. URL http://www.latimes.com/business/la-fi-zestfinance-baidu-20160715-snap-story.html. Retrieved 9/15/2016.

Solon Barocas and Andrew Selbst. Big data's disparate impact. *California Law Review*, 671, 2016. URL https://ssrn.com/abstract=2477899.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, 2016.

Úrsula Hébert-Johnson, Michael P Kim, Omer Reingold, and Guy N Rothblum. Calibration for the (computationally-identifiable) masses. *arXiv preprint arXiv:1711.08513*, 2017.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4066–4076. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6995-counterfactual-fairness.pdf.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226. ACM, 2012.

Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *CoRR*, abs/1609.05807, 2016. URL http://arxiv.org/abs/1609.05807.

Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

Sara Hajian and Josep Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *IEEE transactions on knowledge and data engineering*, 25 (7):1445–1459, 2013.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Proceedings of the 2017 ACM Conference on Innovations in Theoretical Computer Science, Berkeley, CA, USA, 2017*, 2017.

Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. On the (im) possibility of fairness. *arXiv preprint arXiv:1609.07236*, 2016.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1171–1180. International World Wide Web Conferences Steering Committee, 2017.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *arXiv preprint arXiv:1703.00056*, 2017.

Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pages 325–333, 2016.

Matthew Joseph, Jieming Mao, Seth Neel, and Aaron Roth. The role of interactivity in local differential privacy. *CoRR*, abs/1904.03564, 2019. URL http://arxiv.org/abs/1904.03564.

S. Neel, A. Roth, G. Vietri, and Z.S. Wu. Differentially private objective perturbation: Beyond Smoothness and convexity. *ArXiv e-prints*, 2019.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006b.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006c.

Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA*, page 435, 2003. doi: 10.1109/ICDM.2003.1250950. URL https://doi.org/10.1109/ICDM.2003.1250950.

Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

Adam Tauman Kalai, Yishay Mansour, and Elad Verbin. On agnostic boosting and parity learning. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 629–638, 2008. doi: 10.1145/1374376.1374466. URL http://doi.acm.org/10.1145/1374376.1374466.

Giulia C. Fanti, Vasyl Pihur, and Úlfar Erlingsson. Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. *CoRR*, abs/1503.01214, 2015. URL http://arxiv.org/abs/1503.01214.

Andy Greenberg. Apple's 'differential privacy' is about collecting your data—but not your data. *Wired Magazine*, 2016. URL https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/.

Fragkiskos Koufogiannis, Shuo Han, and George J. Pappas. Gradual release of sensitive data under differential privacy. *Journal of Privacy and Confidentiality*, 7, 2017.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014a.

Ryan M Rogers, Aaron Roth, Jonathan Ullman, and Salil Vadhan. Privacy odometers and filters: Pay-as-you-go composition. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*,

pages 1921–1929. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6170-privacy-odometers-and-filters-pay-as-you-go-composition.pdf.

The AMA Team at Laboratoire d'Informatique de Grenoble. Buzz prediction in online social media, 2017. URL http://ama.liglab.fr/resourcestools/datasets/buzz-prediction-in-social-media/.

KDD'99. Kdd cup 1999 data, 1999. URL http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM*, 60(2):12, 2013.

Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. pages 765–774, 2010.

Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 61–70, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4244-7. doi: 10.1109/FOCS.2010.85. URL http://dx.doi.org/10.1109/FOCS.2010.85.

Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. pages 339–356. Springer, 2012.

Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 351–360. ACM, 2013.

Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM Journal on Computing*, 39 (2):606–645, 2009a.

Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM J. Comput.*, 41(6):1558–1590, 2012a. doi: 10.1137/120865094. URL https://doi.org/10.1137/120865094.

Ilias Diakonikolas, Ryan O'Donnell, Rocco A Servedio, and Yi Wu. Hardness results for agnostically learning low-degree polynomial threshold functions. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms*, pages 1590–1606. Society for Industrial and Applied Mathematics, 2011a.

John Ullman and Salil P. Vadhan. Pcps and the hardness of generating private synthetic data. *IACR Theory of Cryptography Conference (TCC)*, 2010. URL http://people.seas.harvard.edu/~salil/research/synthetic-Feb2010.pdf.

Jonathan Ullman. Answering n^2+o(1) counting queries with differential privacy is hard. *SIAM Journal on Computing*, 45(2):473–496, 2016.

Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

Anna Gilbert and Audra McMillan. Property testing for differential privacy. *arXiv preprint arXiv:1806.06427*, 2018.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *IEEE 56th Annual Symposium on Foundations of Computer Science*, 2015.

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private pac learning implies finite littlestone dimension. *arXiv preprint arXiv:1806.00949*, 2018.

Sally A Goldman, Michael J Kearns, and Robert E Schapire. Exact identification of read-once formulas using fixed points of amplification functions. *SIAM Journal on Computing*, 22 (4):705–726, 1993.

Vasilis Syrgkanis, Akshay Krishnamurthy, and Robert E. Schapire. Efficient algorithms for adversarial contextual learning. *CoRR*, abs/1602.02454, 2016. URL http://arxiv.org/abs/1602.02454.

Justin Hsu, Aaron Roth, and Jonathan Ullman. Differential privacy for the analyst via private equilibrium computation. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 341–350, New York, NY, USA, 2013b. ACM. ISBN 978-1-4503-2029-0. doi: 10.1145/2488608.2488651. URL http://doi.acm.org/10.1145/2488608.2488651.

Marco Gaboardi, Emilio Jesús Gallego-Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. *CoRR*, abs/1402.1526, 2014. URL http://arxiv.org/abs/1402.1526.

Elad Hazan and Tomer Koren. The computational power of optimization in online learning. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 128–141, 2016. doi: 10.1145/2897518.2897536. URL http://doi.acm.org/10.1145/2897518.2897536.

Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Cynthia Dwork and Vitaly Feldman. Privacy-preserving prediction. *arXiv preprint arXiv:1803.10266*, 2018.

Raef Bassily, Om Thakkar, and Abhradeep Thakurta. Model-agnostic private learning via stability. *arXiv preprint arXiv:1803.05101*, 2018.

Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 273–282. ACM, 2007.

Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. *SIAM Journal on Computing*, 42(4):1494–1520, 2013.

Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam Smith, and Jonathan Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 775–784. ACM, 2010.

Justin Thaler, Jonathan Ullman, and Salil Vadhan. Faster algorithms for privately releasing marginals. In *International Colloquium on Automata, Languages, and Programming*, pages 810–821. Springer, 2012.

Moritz Hardt, Guy N Rothblum, and Rocco A Servedio. Private data release via learning thresholds. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 168–187. Society for Industrial and Applied Mathematics, 2012.

Vitaly Feldman and Pravesh Kothari. Learning coverage functions and private release of marginals. In *Conference on Learning Theory*, pages 679–702, 2014.

Karthekeyan Chandrasekaran, Justin Thaler, Jonathan Ullman, and Andrew Wan. Faster private release of marginals on small databases. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 387–402. ACM, 2014.

Miroslav Dudík, Nika Haghtalab, Haipeng Luo, Robert E Schapire, Vasilis Syrgkanis, and Jennifer Wortman Vaughan. Oracle-efficient online learning and auction design. In

Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on, pages 528–539. IEEE, 2017.

Alina Beygelzimer, Varsha Dani, Thomas P. Hayes, John Langford, and Bianca Zadrozny. Error limiting reductions between classification tasks. In Luc De Raedt and Stefan Wrobel, editors, *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 49–56. ACM, 2005. doi: 10.1145/1102351.1102358. URL http://doi.acm.org/10.1145/1102351.1102358.

Maria-Florina Balcan, Nikhil Bansal, Alina Beygelzimer, Don Coppersmith, John Langford, and Gregory B. Sorkin. Robust reductions from ranking to classification. *Machine Learning*, 72(1-2):139–153, 2008. doi: 10.1007/s10994-008-5058-6. URL https://doi.org/10.1007/s10994-008-5058-6.

Alina Beygelzimer, Hal Daumé III, John Langford, and Paul Mineiro. Learning reductions that really work. *Proceedings of the IEEE*, 104(1):136–147, 2016. doi: 10.1109/JPROC.2015.2494118. URL https://doi.org/10.1109/JPROC.2015.2494118.

Aharon Ben-Tal, Elad Hazan, Tomer Koren, and Shie Mannor. Oracle-based robust optimization via online learning. *Operations Research*, 63(3):628–638, 2015.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. A reductions approach to fair classification. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *JMLR Workshop and Conference Proceedings*, pages 60–69. JMLR.org, 2018. URL http://proceedings.mlr.press/v80/agarwal18a.html.

Michael J. Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In Jennifer G. Dy and

Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *JMLR Workshop and Conference Proceedings*, pages 2569–2577. JMLR.org, 2018. URL http://proceedings.mlr.press/v80/kearns18a.html.

Daniel Alabi, Nicole Immorlica, and Adam Kalai. Unleashing linear optimizers for group-fair learning and optimization. In *Conference On Learning Theory*, pages 2043–2066, 2018.

Rachel Cummings, Katrina Ligett, Kobbi Nissim, Aaron Roth, and Zhiwei Steven Wu. Adaptive learning with robust generalization guarantees. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 772–814, 2016. URL http://jmlr.org/proceedings/papers/v49/cummings16.html.

Michael J Kearns and Umesh Vazirani. *An introduction to computational learning theory*. MIT press, 1994a.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349 (6248):636–638, 2015b. doi: 10.1126/science.aaa9375. URL http://www.sciencemag.org/content/349/6248/636.abstract.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014b. ISSN 1551-305X. doi: 10.1561/0400000042. URL http://dx.doi.org/10.1561/0400000042.

Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84. ACM, 2007.

ROBIN PEMANTLE and YUVAL PERES. Concentration of lipschitz functionals of determinantal and other strong rayleigh measures. *Combinatorics, Probability and Computing*, 23

(1):140?160, 2014. doi: 10.1017/S0963548313000345.

R. Bardenet and O.-A. Maillard. Concentration inequalities for sampling without replacement. *ArXiv e-prints*, September 2013.

Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory, COLT 1996, Desenzano del Garda, Italy, June 28-July 1, 1996.*, pages 325–332, 1996a. doi: 10.1145/238061.238163. URL http://doi.acm.org/10.1145/238061.238163.

Adam Tauman Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005. doi: 10.1016/j.jcss.2004.10.016. URL https://doi.org/10.1016/j.jcss.2004.10.016.

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 51–60. IEEE, 2010.

Aaron Roth and Adam Smith. Lecture 15: Algorithmic foundations of adaptive data analysis. *https://adaptivedataanalysis.files.wordpress.com/2017/11/lect15.pdf*, 2017.

Jacob Abernethy, Chansoo Lee, Audra McMillan, and Ambuj Tewari. Online learning via differential privacy. *arXiv preprint arXiv:1711.10019*, 2017.

Mihir Bellare, Oded Goldreich, and Erez Petrank. Uniform generation of np-witnesses using an np-oracle. *Information and Computation*, 163(2):510–526, 2000.

Julia Angwin and Hannes Grassegger. Facebook's secret censorship rules protect white men from hate speech but not black children. *Propublica*, 2017. URL https://www.propublica.org/article/facebook-hate-speech-censorship-internal-documents-algorithms.

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. An Empirical Study of Rich Subgroup Fairness for Machine Learning. *ArXiv e-prints*, art. arXiv:1808.08166, August 2018.

Zhe Zhang and Daniel B Neill. Identifying significant predictive bias in classifiers. *arXiv preprint arXiv:1611.08292*, 2016.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, and John Langford. A reductions approach to fair classification. *Fairness, Accountability, and Transparency in Machine Learning (FATML)*, 2017. URL http://fatml.mysociety.org/media/documents/reductions_approach_to_fair_classification.pdf.

Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. *arXiv preprint arXiv:1702.06081*, 2017.

Michael J Kearns and Umesh Virkumar Vazirani. *An Introduction to Computational Learning Theory*. MIT press, 1994b.

Ilias Diakonikolas, Ryan O'Donnell, Rocco A. Servedio, and Yi Wu. Hardness results for agnostically learning low-degree polynomial threshold functions. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1590–1606, 2011b. doi: 10.1137/1.9781611973082.123. URL https://doi.org/10.1137/1.9781611973082.123.

Maurice Sion. On general minimax theorems. *Pacific J. Math.*, 8(1):171–176, 1958. URL https://projecteuclid.org:443/euclid.pjm/1103040253.

George W. Brown. Some notes on computation of games solutions, Jan 1949. URL https://www.rand.org/pubs/research_memoranda/RM125.html.

Julia Robinson. An iterative method of solving a game. *Annals of Mathematics*, pages 10–2307, 1951.

Constantinos Daskalakis and Qinxuan Pan. A counter-example to Karlin's strong conjecture for fictitious play. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 11–20. IEEE, 2014.

Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing*, 41(6):1558–1590, 2012b.

Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM Journal on Computing*, 39 (2):606–645, 2009b.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

Michael Kim, Omer Reingold, and Guy Rothblum. Fairness through computationally-bounded awareness. In *Advances in Neural Information Processing Systems*, pages 4842–4852, 2018.

Stephen Gillen, Christopher Jung, Michael Kearns, and Aaron Roth. Online learning with an unknown fairness metric. In *Advances in Neural Information Processing Systems*, pages 2600–2609, 2018.

C Ilvento. Metric learning for individual fairness. Manuscript submitted for publication, 2019.

Guy N Rothblum and Gal Yona. Probably approximately metric-fair learning. *arXiv preprint arXiv:1803.03242*, 2018.

Maurice Sion et al. On general minimax theorems. *Pacific Journal of mathematics*, 8(1): 171–176, 1958.

Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *COLT*, volume 96, pages 325–332. Citeseer, 1996b.

Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132:1–63, 1997.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 2003.

Michael J Kearns and Umesh Virkumar Vazirani. *An introduction to computational learning theory*. MIT press, 1994c.

Robin Pemantle and Yuval Peres. Concentration of lipschitz functionals of determinantal and other strong rayleigh measures. *Combinatorics, Probability and Computing*, 23(1): 140–160, 2014.

Seth Neel, Aaron Roth, and Zhiwei Steven Wu. How to use heuristics for differential privacy. *arXiv preprint arXiv:1811.07765*, 2018.

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Ann. Statist.*, 28(5):1302–1338, 10 2000. doi: 10.1214/aos/1015957395. URL https://doi.org/10.1214/aos/1015957395.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.